Image Interpolation with Hidden Markov Model

IMAGE INTERPOLATION WITH HIDDEN MARKOV MODEL

BY

AMIN BEHNAD, B.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

Master of Applied Science (2010)
(Electrical & Computer Engineering)

McMaster University
Hamilton, Ontario, Canada

TITLE:	Image Interpolation with Hidden Markov Model
AUTHOR:	Amin Behnad B.Sc., (Electrical Engineering) Sharif University of Technology, Tehran, Iran
SUPERVISOR:	Prof. Xiaolin Wu
NUMBER OF PAGES:	xi, 63

To my parents

Abstract

The first part of this thesis is concerned with efficient adaptive image interpolation techniques for real-time applications. A new image interpolation algorithm is developed that combines optimal data fusion and context modeling of images. Specifically, two estimates of missing pixels obtained by cubic interpolation in perpendicular directions are optimally fused under minimum mean square (MMSE) criterion. The fused result is further improved by a context-based error feedback mechanism to compensate for the error of cubic interpolation. The proposed image interpolation algorithm preserves edge structures well and achieves superior visual quality. This is accomplished at low computational complexity, making the new algorithm suitable for hardware implementation.

The main part of this thesis is devoted to a more sophisticated image interpolation algorithm based on hidden Markov modeling (HMM). Most of existing interpolation algorithms rely on point by point decisions to estimate the missing pixels. In contrast, the HMM approach of image interpolation estimates a block of missing pixels via maximum a posterior (MAP) sequence estimation. The hidden Markov model can incorporate the statistics of high resolution images into the interpolation process and the MAP estimation technique can exploit high-order statistical dependency between pixels. The proposed HMM-based image interpolation algorithm is implemented and

its performance is evaluated and compared with existing methods. The comparison study shows that the HMM-based image interpolation algorithm can reproduce cleaner and sharper image details than its predecessors, while suppressing common interpolation artifacts such as ringing, jaggies, and blurring.

Acknowledgements

I would like to give my special thanks to my supervisor Professor Xiaolin Wu, for his suggestions and support. I also want to appreciate his kind personality. He definitely has played a great role in my life.

My deepest gratitude goes to my parents, my brother, Aydin, my sister, Ana, and my brother in law, Ali Rokoui.

I would also thank all my friends that helped me in my studying and enjoying my life here: Xiangjun Zhang, Xiaohan Wang, Amin Abdossalami, Reza Pournaghi, Behzad Mahdavikhah, Navid Samavati, Ramin Mafi, Saba Moghimi, Bahram Marami, Mahdi Nabaee, Sina Nia-Kosari, Navid Shahdi, Mohammad-Reza Dadkhah, Hossein Kassiri, Fabiola De Vierna, Ali-Reza Shoa, Kimia Ghobadi, Mohammad-Mehdi Farazmand and Sahar Alipour.

Contents

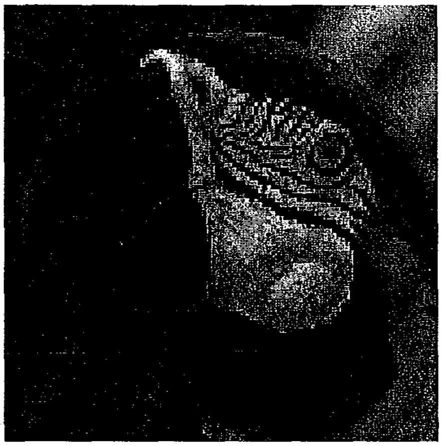
List of Figures

Chapter 1

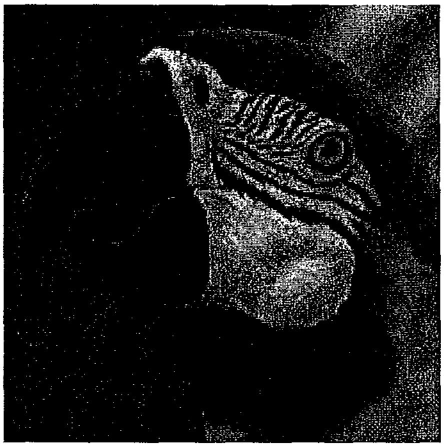
Introduction

The ability of human visual system to resolve objects in a captured digital image depends on many factors, among which the spatial resolution is of great importance. It largely determines the perceptual quality of a digital image. The higher the spatial resolution of an image, the better reproduction of fine structures in a captured image. To demonstrate the relation of spatial resolution on image quality, the reader is invited to compare two images of one scene but of different spatial resolutions in Fig. 1.1.

For many applications the native resolution of a captured image is lower than required and should be enhanced by computer software. Image interpolation is the technique to compensate for the insufficiency of the native resolution of an image acquisition device. In general, a digital signal is obtained by sampling a continues signal at a specific sampling rate. Recovering the original continuous light field from corresponding discrete samples (pixels) is the ultimate goal of image interpolation. Image interpolation is a classical problem of image processing, dated back to the very beginning of the field. It has a wide range of applications spanning from consumer electronics to visual arts and medical imaging. Some of these applications are listed



(a)



(b)

Figure 1.1: Images with different resolutions. (a)Low resolution (LR) and (b)High resolution (HR).

as follows.

A direct application of image interpolation is large format reproduction of an image, such as those in high quality large prints in magazines, catalogs or wall posters. In these cases, the resolution of printers is much higher than that of a digital camera, and hence one has to convert the spatial resolution of the input image. Similarly, image interpolation is necessary to prepare standard definition images or videos for displaying on high-definition monitors, such as large HDTV panels.

In remote sensing, users want to resolve as much detail on the ground as possible, say identifying a street sign. But the highest resolution that a satellite imager can achieve is only 40 centimeter, obviously not sufficient for the above purpose. Here image interpolation technique can be used to improve the native resolution of the satellite image.

In some cases the image acquisition process itself may be harmful to the object being imaged, and the degree of harm is proportional to the spatial resolution of the imager. For instance, when conducting a CT (Computed Tomography) scan of a patient in medical imaging, a sophisticated image interpolation technique can be beneficial to the patient because it can reduce the dosage of X-ray without sacrificing the image resolution.

The other important application of image interpolation is video deinterlacing. In digital camcorders frames are recorded as a sequence of fields, each field consists of alternatively even or odd lines of a frame. For displays, pairs of fields need to be merged into frames. The process of converting the interlaced frames to progressive frames is called deinterlacing, which poses a particular type of image interpolation problem. A detailed study on deinterlacing is presented in Chapter 2.

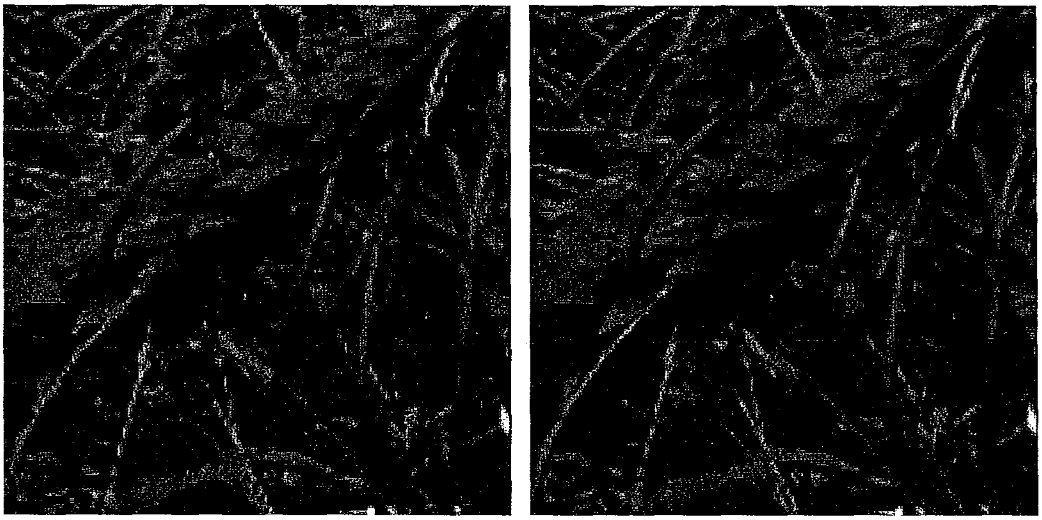
For the majority of digital cameras and camcorders, each pixel sensor can only capture a single color and hence the output is a spatially down sampled color image in a specific mosaic pattern. Demosaicking process that reconstructs a full image from the mosaic data is also a problem of image interpolation.

Video frame rate upconversion (FRUC), is another application of image interpolation. FRUC is the process to reconstruct the missing or skipped frames, which is a problem of temporal interpolation between neighboring video frames. It is the necessary technology to reproduce the video in slow motion or to prevent jerkiness of fast motions in video playback.

Image interpolation has a role to play in computer vision and robotics as well. For example, in motion analysis, motion vectors of subpixel precision are required. The subpixel precision can only be achieved through image interpolation.

For the numerous applications as identified above and beyond, and to satisfy users' insatiable appetite for details in images, image interpolation has remained an active research topic in both academia and industry.

The technical challenge of image interpolation is to reconstruct the high-frequency components or fine textures. Based on Nyquist-Shannon sampling theorem [4], continuous signals with limited frequency bandwidth are recoverable at specific sampling rate. However around sharp edges and boundaries of objects, due to rapid changes of the light field, the 2D image waveform has unlimited bandwidth and hence mathematically it is impossible to completely reconstruct the original sampled signal with a limited number of pixels. To aggravate the difficulty of reproducing the edges, the human visual system is highly sensitive to edges which signify attributes of an object (e.g., shape, boundaries, surface and other visual characteristics). Thus, even small interpolation errors around edges are highly visible and thus drastically degrade the visual quality of entire image. As such, the performance of image interpolation techniques is largely determined by how well they preserve edges and fine structures of an image (Fig. 1.2). Most image interpolation algorithms try to fit the 2D image waveform to a mathematical model and use the model to estimate the high-frequency components of an image. In the next section, a number of these algorithms, including those of the current state of the art, are reviewed.



(a) (b)

Figure 1.2: Reproduction qualities of different interpolators. (a) Bicubic and (b) Fast Directional Interpolation (FDI).

1.1 Literature Review

Interpolation is a classic mathematic problem and has a long history. The first reported application of interpolation dates back to 300 BC when linear interpolation was used by Babylonian astronomers. In 600 AD Chinese mathematician Liu Zhuo used an equivalent of order-two Newton interpolation to do the computations of "Imperial Standard Calendar". Indian astronomers in 625 AD also applied similar interpolation techniques [5].

The first application of interpolation for digital images was reported in early 1970's. Since then, many two-dimensional interpolation methods have been proposed, varying from extension of previous (1D) interpolators to 2D space, to sophisticated interpolation algorithms specially designed for digital images.

Developed interpolation methods are of different tradeoffs between resulting visual quality and computational/time efficiency. For the applications requiring high speed, the common solutions are nearest neighbor pixel replicating or simple image-independent linear filters, such as bilinear interpolator and bicubic interpolator [6]. These filters are linear since each estimated pixel is formulated as a linear combination of adjacent pixels with fixed weights. Design of such filters is based on the interpolation kernel they use. Theoretically, to reconstruct a bandlimited signal $f(x)$ sampled at a rate higher than Nyquist sampling rate ($\frac{1}{T}$), the best kernel is the sinc function ($\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$) and the reconstructed signal $\hat{f}(x)$ can be formulated as

$$\hat{f}(x) = \sum_{n=-\infty}^{\infty} f(nT)\text{sinc}(\frac{x - nT}{T}) \tag{1.1}$$

However, as described in previous section, image signal around high frequency component is not bandlimited. Furthermore the sinc function is not compactly supported (i.e. unbounded in time domain). Hence in real application it is not feasible to use this function. The sinc function in (1.1) can be replaced by any other interpolation kernel. Key in [6] proposed the cubic convolution for image interpolation exploiting larger window and a sinc-like kernel to improve the previous linear filters, at the cost of higher complexity. This method is used in image processing software e.g. Adobe Photoshop. Other interpolation kernels were also proposed and a number of them were compared by Lehmann and Spizter [7].

However these simple linear filters are isotropic and ill suited for directional image waveforms, and they tend to produce severe interpolation artifacts in areas of edges and fine textures. Aliasing, ringing and blocking are common artifacts visible in the output of these interpolators. Aliasing occurs due to aliased frequency spectrum of sampled signal and loss of high frequency components. Loss of texture is the perceptual effect of aliasing [8]. Ringing usually happens when the interpolation kernel has oscillation (Gibbs phenomena [4]). Another drawback of interpolation with these windowed kernels is blocking effect which is due to finite influence of any pixel on neighboring pixels.

To improve the visual quality of above said interpolators, researchers tried to enhance these filters. In [9], Liang proposed the idea of blending kernels by adopting affine weights to combine interpolation kernels. The interpolation scheme of El-Kbamy *et al.*, applies adaptive coefficients on cubic convolution to better match the structure of local contexts and improve the visuality of enlarged image [10]. Ramponi in [11] proposed to modify the Euclidean distances by a warped distance-based adaptive image interpolation method. This method is applicable to linear filters like bilinear and cubic convolution to enhance their performance. In [12], the authors applied a postprocessing algorithm improving perceptual quality of the interpolated images. The intensities of pixels around edges were modified to reduce ringing artifacts caused by non-adaptive interpolation methods. Hwang and Lee proposed an inverse gradient to the structure of conventional bilinear and bicubic interpolation to sharpen edges [13]. In [14], splines are applied as image model to interpolate the high resolution images from nonuniformly sampled images. The splines were used for filtering and an adaptive smoothness was used as a regularization constraint.

Although these algorithms provide better objective (PSNR) results, they are susceptible to visual defects due to their weak adaptability to varying structures across a natural image.

In pursuit of better visual quality and higher computational efficiency, more complex and adaptive interpolation algorithms were introduced. Among these methods, edge-guided schemes are of interest because human visual system is highly sensitive to edge structures, which convey much of the image semantics. Hence for most of applications the performance of an image resolution upconversion algorithm is judged by how well it can recover high frequency components of the underlying continuous light field. Different from the task of upsampling a one-dimensional signal, upsampling an image can and should exploit the fact that high frequency image features, such as edges and textures, are anisotropic in nature. The spectrum of the edges is also asymmetric since the frequency is low along the edge-directions and high in the perpendicular direction.

Edge-directed image interpolation methods can generally be divided in two classes. In the first class, called explicit edge-directed image interpolation, first the edges in the image are detected and interpolation is performed along their directions (edge directions are of low-frequency and hence recoverable by directional interpolation). In the second class of edge-directed interpolation methods, edge information is implicitly exploited in interpolation. The reproduction quality of interpolated images for both of these types of methods mainly depends on faithfully detecting the edge directions. Otherwise the penalty of filtering in wrong directions is high both subjectively and objectively.

Wong and Alleback presented an edge-directed image interpolator in [15]. The method is based on iteratively detecting the edge map of high resolution image, rendering which is bilinear interpolation for the non-edge area and correction by measuring the disparity between original low-resolution samples and the ones predicted by the results of rendering phase.

Another edge guided interpolator is proposed in [16]. The method first finds the texture orientation map using directional Gabor filter in subpixel precision, then the information training set is also applied by kernel Fisher discriminant to further refine the interpolation direction.

The method published by Carrato and Tenze [17] replicates the pixels and then corrects the values by a rational operator, designed to produce least average of squared error for some predefined edge patterns. Their method is simple and suitable for hardware implementations.

Dube and Hong published a simple classification-based interpolation scheme [18], that first classifies the edges into some predetermined prototypes of directions and then applies suitable directional filtering. Zhang and Wu also proposed a classification-based image interpolation technique [16].

Jensen and Anastassiou in [2] proposed an operator to fit detected edges by some templates to improve the visual quality of interpolated image. In some cases the reproduction quality of [2] is better than linear filters and reconstructed edges are sharper. However, the edges in the output images of [2] sometimes look contrived and unnatural. Since the large-scale context of edges was not considered, small variations in texture could be interpreted as an edge and mistakenly fitted by an edge. Such mistakes result in distortions of textures in such cases. The time complexity of this algorithm is also too high for real-time applications.

In [19] Li and Orchard proposed a Wiener filtering like-algorithm called new edge-guided image interpolation (NEDI). NEDI interpolates missing pixels based on the covariance of HR image computed from LR samples and in a geometric duality principle. Geometric duality principle means that the covariance of the HR image and LR image are similar. The main drawback of this algorithm is high computational complexity. Although they proposed to perform their algorithm on active areas of an image and rely on bilinear interpolation for other parts, their algorithm is still too complex for real-time applications. Besides, in the area with small curvatures and several intersections, the NEDI algorithm suffers from speckle noises and artifacts. However, this method is one of the best interpolation schemes regarding visual quality. As such in this thesis we use NEDI as a benchmark for subjective evaluation of interpolation results.

Li and Nguyen published an interpolation scheme that converts edge information to a vector of given weights to different directions. In this scheme, the interpolated image is related to the minimal energy state of a 2D random Markov field given the low-resolution image [20]. But the random Markov field methods are iterative and very time consuming.

Wavelets were also a useful tool for image interpolation [21], [22], [23] and [24]. In this type of methods some features of high-resolution image are predicted from low-resolution observations. In [21], Carey et al. proposed a wavelet-based interpolation method that preserves some local regularity measurements in the up-scaled image.

In [25], Zhang and Wu proposed to fit the image to a 2D autoregressive (AR) model. In this algorithm each missing pixel in a block is formulated as a linear combination of four known neighboring pixels. The associated coefficients (AR parameters) and a block of missing pixels are estimated jointly. This method is among the best methods in term of objective assessment and also subjective quality. However this method needs to solve a nonlinear optimization problem for each block and thus it is unsuited for real-time applications.

In the next section the proposed interpolation algorithms and contributions of this thesis are summarized.

1.2 Thesis Contributions

In this work, two new image interpolation techniques are presented. Both techniques belong to the class of edge-guided image interpolation methods. But they differ in algorithmic approach and the trade-off between complexity and performance. One of them is designed as a fast adaptive directional image interpolator (FDI), which is a modification of the edge-guided interpolation method proposed in [1]; the other is an edge-directed interpolation method based on hidden Markov model (HMM-DI). The former adopts a pixel-by-pixel estimation scheme, whereas the latter casts and solves image interpolation as a problem of optimal sequence estimation. Both of these methods are tested on different scenes and merits and limits of each are compared with some of the existing methods.

1.2.1 Fast Directional Interpolation

This image interpolation technique is a modification and improvement of an earlier work [1]. In [1], Zhang and Wu proposed a linear minimum mean square method of fusing two directional interpolators. In this technique, the two interpolation errors were assumed to be statistically independent. In this study, however, we find that for natural images significant correlations exist between different directional estimates, particularly in areas of strong edges. We remove the incorrect assumption of [1] in the design of directional interpolators and propose a new fusion-based directional interpolation (FDI) technique that accounts for statistical dependencies between different directional estimates. A context-based error-feedback mechanism is also proposed for further improvement of the interpolation results.

This method preserves edges and fine textures of images very well. It is also designed to be fast and simple, and hence suitable as real-time hardware solution for image/video interpolation. The proposed fast directional interpolation (FDI) algorithm has a general framework. In this work we apply FDI to 2D image resolution upconversion and video deinterlacing.

1.2.2 Interpolation with Hidden Markov Model

There are two issues with all of the reviewed edge-guided interpolation algorithms. First, the edge direction is estimated from the low-resolution input image, and it is error prone when insufficient sampling rate causes aliasing. Second, the edge direction is determined and directional interpolation carried out on a pixel by pixel basis disregarding the spatial coherence of edge points.

In this thesis we address these issues and overcome the above said drawbacks by applying the hidden Markov model (HMM) technique to solve the problem of image resolution upconversion. The hidden states of the HMM correspond to edge directions or smooth waveform of the underlying high-resolution image, which are not directly observable in the low-resolution image. Each of these HMM states is associated with an interpolator that suits the corresponding waveform. Ideally, if the HMM state was known for each pixel the corresponding interpolator would be applied for best performance. As such, the interpolation problem is converted to one of estimating the hidden states based on the observed low-resolution image, and it is solved by a maximum a posterior (MAP) sequence estimation method. The advantages of our HMM-based interpolation approach are two folds:

1. it facilitates the incorporation of the statistics of high-resolution images of a training set into the interpolation process;

2. it makes a joint decision, via MAP sequence estimation, on interpolated pixels rather than one pixel at a time in isolation.

The comparison study verifies that the proposed HMM-based image interpolation technique achieves superior visual-quality and removes common artifacts such as ringing, jaggies, and blurring.

1.2.3 Organization

The remainder of this thesis is organized as follows. Chapter 2 presents the fusion-based image interpolation algorithm and its applications. Chapter 3 proposes our new interpolation method based on hidden Markov modeling. Conclusion remarks and suggestions for future works are presented in the Chapter 4.

Chapter 2

Fast Adaptive Directional Interpolation of Images

In this chapter our first adaptive image interpolation algorithm is presented, called Fast Directional Interpolation (FDI). In this method, estimates of a missing pixel in multiple directions are fused under proposed optimal data fusion, based on minimum mean square error (MMSE) estimation. A similar approach for video deinterlacing is also presented in this chapter.

2.1 Overview

In this thesis the LR image is considered as an ideal Dirac down-sampled result of the original HR image. The formation of high-resolution (HR) image from low-resolution samples is shown in the Fig. 2.1.

The main goal of image interpolation is to estimate the values of pixels depicted by white circles in the Fig. 2.1. In the regions of smoothness, where there is not a sharp

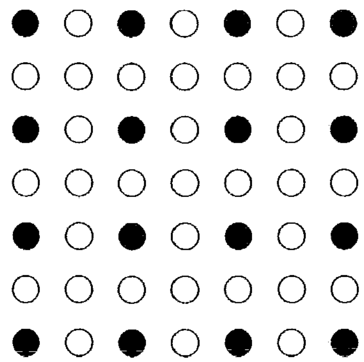


Figure 2.1: Formation of a high-resolution image from low-resolution samples. The white dots are interpolated pixels and black dots are original pixels.

edge or rapidly varying texture, the task of interpolation can be easily performed since the rate of sampling is fairly higher than Nyquist-Shannon sampling limit. The main challenge of interpolating the missing pixels is around the edges as described in Chapter 1. Since usually there is a dominant direction in these area, isotropic filters are error prone and produce blurred edges.

One idea to handle the difficulty reproducing the HR image while keeping the edge/texture orientation is implicitly apply the edge information by estimating the missing pixels via fusing interpolated values in some directions of different angles as proposed in [1]. However, there are two issues with this method:

1. In this technique, the two interpolation errors were assumed to have zero correlation. We experimentally show this assumption is not correct and propose a new fusion-based directional interpolation (FDI) technique that accounts for statistical dependencies between different directional estimates.

2. In this paper only two orthogonal directions are considered for estimating each unknown pixel. Hence, when the edge directions are too different from these directions the interpolator fails to reconstruct them properly as depicted in

Fig. 2.2. In FDI we added some more directions as the safe guard to handle these cases.

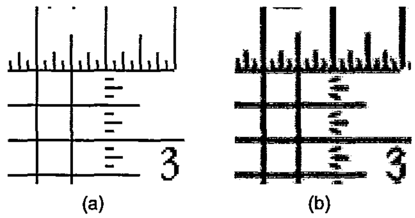


Figure 2.2: Weakness of method in [1] to reproduce the sharp horizontal and vertical lines, (a) is the original image and (b) reconstructed by the method of [1].

The rest of this chapter is organized as follows. The formulation of general data fusion under minimum mean square error (MMSE) is proposed in Sec. 2.4.1. The detail of FDI algorithm for image resolution upconversion is described in Sec. 2.3. A deinterlacing method based on FDI is also proposed in Sec. 2.4. The comparison between results of FDI and some of the other existing methods is presented in Sec. 2.5. The chapter closes with some conclusion remarks in Sec. 2.6.

2.2 Optimal Data Fusion of Multiple Dependent Estimates

Data fusion is a statistical inference methodology of combining two or more estimates of an unknown variable via a function to produce a more robust estimate. For the interest of this study on adaptive directional image interpolation, we consider n estimates $d_1, d_2, ..., d_n$ of missing pixel y that are produced by n different directional interpolators. For low complexity and ease of implementation we adopt affine weights $\omega_1, \omega_2, ..., \omega_n$ to linearly fuse $d_1, d_2, ..., d_n$. These weights are optimized to minimize the expected estimation error, namely

$$\min_{\omega_1,\omega_2,\ldots,\omega_n} E(|y - \sum_{i=1}^{n} \omega_i d_i|^2) \tag{2.1}$$

subject to:

$$\sum_{i=1}^{n} \omega_i = 1 \tag{2.2}$$

Assuming each of n estimates d_i's has the error e_i ($d_i = y + e_i$), the solution of the above optimization is

$$\boldsymbol{\omega} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\theta} \tag{2.3}$$

where

$$\boldsymbol{\omega} = (\omega_1, \omega_2, ..., \omega_{n-1}) \tag{2.4}$$

$$\boldsymbol{\Sigma}_{i,j} = \begin{cases} \sigma_i^2 + \sigma_n^2 - 2\sigma_{i,n}, & i = j; \\ \sigma_n^2 - \sigma_{i,j} - \sigma_{i,n}, & i \neq j. \end{cases} \tag{2.5}$$

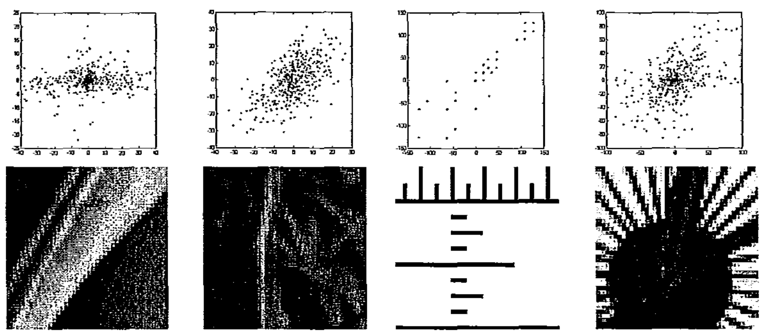
$$\boldsymbol{\theta} = (\sigma_n^2 - \sigma_{1,n}, \sigma_n^2 - \sigma_{2,n}, ..., \sigma_n^2 - \sigma_{n-1,n}) \tag{2.6}$$

with σ_i^2's being variances of individual estimation errors e_i, and $\sigma_{i,j}$'s covariances between variables e_i and e_j. For $n > 2$, the complete solution of (2.3) may be too expensive for real-time applications. But for $n = 2$, we can compute the optimal weights efficiently

$$\begin{aligned} \omega_1 &= \frac{\sigma_2^2 - \sigma_{1,2}}{\sigma_1^2 + \sigma_2^2 - 2\sigma_{1,2}} \\ \omega_2 &= \frac{\sigma_1^2 - \sigma_{1,2}}{\sigma_1^2 + \sigma_2^2 - 2\sigma_{1,2}} \end{aligned} \tag{2.7}$$

For this reason in case of three or more estimates we will choose and fuse the two estimates whose variances are the two smallest among $n > 2$ estimates.

Zhang and Wu proposed the idea of fusing the results of two mutually orthogonal directional interpolators [1]. But they ignored the term $\sigma_{1,2}$ in (2.7) by assuming that the two interpolation errors are statistically independent. But their assumption is an oversimplification for many types of natural images. In Fig. 2.3, four test images are presented together with the corresponding joint distributions of the two directional interpolation errors in 45 and -45 degrees. In cases (b) through (d) the covariance $\sigma_{1,2}$ is significant, and it cannot be dropped from the weights ω_1 and ω_2 without degrading the interpolation performance.



(a)0.17 (b)0.49 (c)0.71 (d)0.43

Figure 2.3: Joint distributions of directional interpolation errors in two diagonal directions. The associated numbers are correlation coefficients.

To estimate the statistics of error terms in (2.7) we take the following approach. Assuming the image signal is stationary in the locality of the missing pixel, we first estimate the unknown pixels along each direction, then interpolate the neighboring known pixels ($Y(i,j)$) by estimates of the unknown pixels along the same direction ($\hat{Y}(i,j)$) and compare them with existing known pixels to measure the errors and the values of variances and covariances

$$\sigma_k^2 = \frac{1}{N-1} \sum_{i,j \in W} (Y(i,j) - \hat{Y}_k(i,j))^2 \quad k = 1,2 \tag{2.8}$$

and

$$\sigma_{1,2} = \frac{1}{N-1} \sum_{i,j \in W} (Y(i,j) - \hat{Y}_1(i,j))(Y(i,j) - \hat{Y}_2(i,j)) \tag{2.9}$$

where the W is the local window centered at the estimating pixel and N is the number of known pixels in W.

The FDI technique outlined above has the same level of complexity as the technique of [1]. Since each pixel is processed in the same way and the process only involves linear filtering operations, FDI lends itself conveniently to parallel and hardware implementations.

2.3 FDI for Image Resolution Upconversion

In this section we reexamine the problem of upconverting the spatial resolution of an image via interpolation.

To make full use of low resolution image data and in the interest of low complexity, we choose four directions in which the low resolution pixels are aligned, and two of which are mutually orthogonal. Fig. 2.4 depicts these four interpolation directions. In the presence of sharp horizontal and vertical edges, the fused diagonal interpolation result may be very poor. We resort to the bicubic filter designed to act on pixels in horizontal and vertical directions (Fig.2.4), whenever the error variances in diagonal directions are much larger than an empirically estimated threshold.

As mentioned above, the applied interpolator for our design is bicubic filter. In this method, the intensity assigned to each missing pixel is modeled as a function of

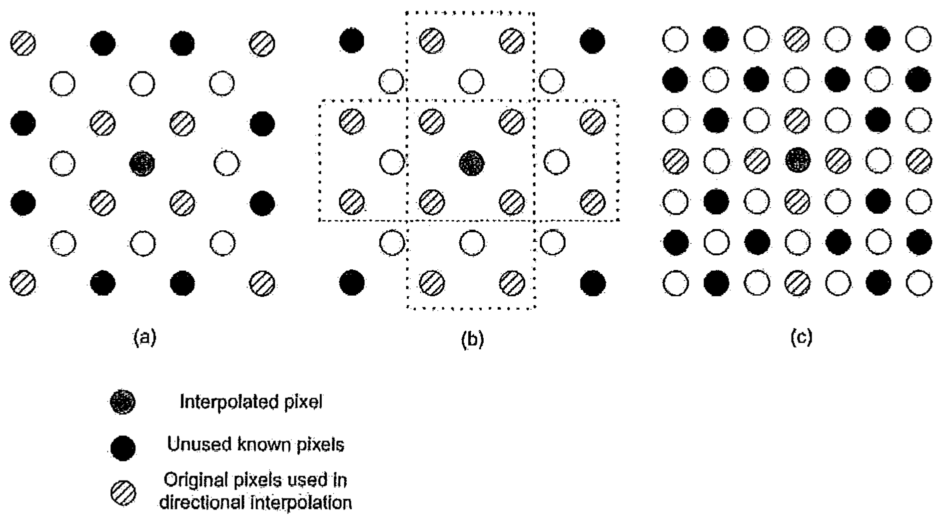


Figure 2.4: (a),(b) Interpolation directions for the first step of proposed algorithm. (c) Directions for interpolating remaining pixels by pre-estimated values.

its sixteen nearest neighbors as

$$I(x,y)=\sum_{i=0}^{3}\sum_{j=0}^{3}a_{ij}x^iy^j \tag{2.10}$$

The coefficients are determined by solving the sixteen equations obtained from the sixteen nearest pixels. For the horizontal and vertical directions of Fig. 2.4 this formulation will be changed to

$$I(x,y)=\sum_{i=0}^{3}\sum_{j=0}^{1}a_{ij}x^iy^j \tag{2.11}$$

and

$$I(x,y)=\sum_{i=0}^{1}\sum_{j=0}^{3}a_{ij}x^iy^j \tag{2.12}$$

As such, the coefficients of the applied filters will be as depicted in Fig. 2.5.

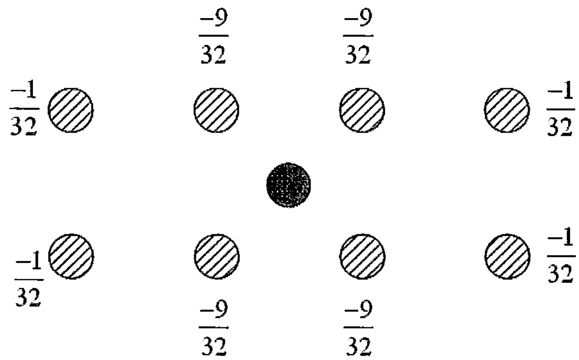


Figure 2.5: Axial cubic filter applied in proposed FDI algorithm.

2.3.1 Noise Effect

The input LR image may contain noises from various sources: sensor noises, quantization errors in compression, etc. The compound noise of all these sources can be modeled as white Gaussian of variance σ_n^2. In smooth areas the interpolation errors in both directions are very small. This may make the solution of (2.7) unstable. To avoid this, the optimization problem (2.1) should be reformulated to factor in the image noise

$$\min_{\omega} E(|y - \omega(d_1 + n_1) - (1 - \omega)(d_2 + n_2)|^2) \tag{2.13}$$

and the optimal weighting will be

$$\omega = \frac{\sigma_2^2 - \sigma_{1,2} + 0.64\sigma_n^2}{\sigma_1^2 + \sigma_2^2 - 2\sigma_{1,2} + 1.28\sigma_n^2} \tag{2.14}$$

Since the noise is convolved with a cubic filter its variance is reduced to $0.64\sigma_n^2$. In the smooth area the terms σ_1, σ_2 and $\sigma_{1,2}$ are negligible. Therefore, the optimal weight converges to 0.5. This modification improves the subjective and objective quality of the proposed interpolation method substantially.

2.3.2 Error Compensation

After finding the corresponding weights, each missing pixel is found as

$$Y(i,j) = \omega \hat{Y}_1(i,j) + (1-\omega)\hat{Y}_2(i,j) \tag{2.15}$$

where $\hat{Y}_1$ and $\hat{Y}_2$ are two estimates in perpendicular directions. As such the expected error of final estimate is

$$E\{e\} = \omega E\{e_1\} + (1-\omega)E\{e_2\} \tag{2.16}$$

In many regions of an image, cubic interpolator suffer from large error and hence the interpolated values by FDI are susceptible to large errors. To handle this problem, we take a classification approach to reduce the error of cubic interpolator.

In theory it is impossible to exactly detect and compensate the error of interpolation:

Interpolation Error Theorem- *Let $I(x)$ be a real function (N+1)-times differentiable in the interval partitioned as $[a = x_0, x_1, ..., x_N = b]$. Assume $I(x)$ is interpolated with a N^{th}-order polynomial $P_N(x)$. Then for every $\hat{x} \in [a,b]$ there is a point $\xi \in [a,b]$ where:*

$$I(\hat{x}) = P_N(\hat{x}) + \frac{I^{(N+1)}(\xi)}{(N+1)!}\prod_{k=0}^{N}(\hat{x} - x_k) \tag{2.17}$$

where ξ is independent of $\hat{x}$. [26]

In general it is impossible to exactly determine ξ when, similar to pixel intensity function, the interpolated signal is unknown. However we will develop a new classification-based approach to reduce the interpolation error.

Fitting the signal waveform by a smooth third degree polynomial, cubic interpolation often has the largest error around sharp edges where the image light field is not continuous. Based on this, we detect and classify possible cases and measure the error of cubic interpolator in training for each. The key to success for this algorithm is the formation of context (set of pixels we utilize to estimate missing pixel) to characterize the image signal waveform. The feature vector we defined for a pixel being interpolated by a set of LR pixels $x_1, \cdots, x_4$ is $\vec{\phi} = (x_1 - x_2, x_2 - x_3, x_3 - x_4)$. The magnitude of differences obey exponential distribution and hence are quantized in 8 levels with exponentially increasing step size. Including the signs, we have $17^3 = 4913$ classes. To find the table of errors, an appropriate set of training images is used. Given a context, the error of cubic interpolator is computed by comparing with existing HR pixels and the mean value of these errors for each class is saved in a look-up-table. This process is time consuming, but is performed off line and in interpolation process first the context is computed and then the error of cubic interpolation is computed and compensated from interpolation results.

This technique is not restricted to cubic convolution, but suitable for any interpolation kernel. This method improves visual quality and increases the PSNR. Error compensation can be interpreted as de-noising, since it reduces the interpolation error by applying prior knowledge from training data.

2.3.3 Algorithm Implementation

The algorithm of FDI for image resolution upconversion consists of two passes. First, the set of unknown pixels pictured by white circles in Fig. 2.4 are estimated by cubic interpolator. For this set of pixels in all of four pre-defined directions, original samples of LR image are available. First the error-variances of diagonal directions are computed and if they are larger than a pre-determined threshold (th), then horizontal and vertical estimates are fused. Otherwise, the diagonal directions are used.

Once these missing pixels are estimated, in the second pass, the remaining pixels in Fig. 2.4 (grey circles) are estimated via fusing only horizontal and vertical interpolated values since for this configurations there are not enough samples in other directions. As depicted in Fig. 2.4 one of these directions consists of original samples and the other consists of estimated values of the first pass.

The implementation of error-compensated FDI (EC-FDI) is identical but the context-based error compensation is applied when estimates in diagonal and axial directions are computed. The verification and fusion is similar to FDI.

Although the interpolation algorithm based on FDI is described to double the size of images, it is applicable for any scaling factor $S = 2^m$, where m is an integer. For scaling factors which are not a power of two ($2^m < K < 2^{m+1}$), first FDI is used to enlarge it 2^m times and then we may expand it by a linear filter e.g. cubic convolution k times such that: $k2^m = K$.

2.4 FDI for Video Deinterlacing

2.4.1 Overview

All of analog and many digital camcorders scan and save frames as consecutive fields. Each field consists of alternatively odd and even lines of corresponding frame. This process is called interlacing and the process of converting interlaced frames to progressive sequence of frames is called deinterlacing which is necessary in displays. Many LCD's and plasma TV sets have built-in circuitry to carry out this task.

Deinterlacing methods are generally classified in intra-field and inter-field categories. In former method each field is processed separately to reproduce the entire field and in the latter the between-frame correlations and motion-compensation algorithms are considered to improve the visual quality of resulting frame. However, beside much higher complexity, in case with incorrect provided motion-estimation information they fail to yield good results.

Basically, intra-field video deinterlacing is a problem of 2D image interpolation with a particular spatial configuration (Fig. 2.6). Each missing line is to be estimated by pixels of neighboring lines. Line-doubling (duplicating each line) and line-averaging are the simplest deinterlacing methods. But identical to isotropic filters, they tend to introduce visual artifacts and therefore researchers proposed more sophisticated algorithms, faithfully applying available pixels to interpolate missing data.

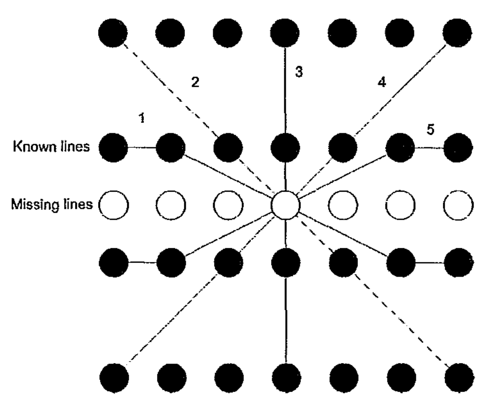


Figure 2.6: Directions used for proposed de-interlacing method.

Similar to resolution upconversion problem, many directions can be considered to estimate the unknown pixels. This implies that we can take the similar approaches as edge-guided image interpolation methods to find the directions of smoothness. The conventional edge-based line averaging (ELA) is a simple edge-directed method [27]. In ELA, the best direction is figured out by computing the absolute value of differences in three (or more) directions showed in Fig. 2.6 and then averaging is applied in that direction:

$$Y(i,j) = \frac{1}{2}(X(i-1,j-\hat{s}) + X(i+1,j+\hat{s})) \tag{2.18}$$

where

$$\hat{s} = \arg \min_{-1\leq s\leq 1} \left|X(i-1,j-s) - X(i+1,j+s)\right| \tag{2.19}$$

Based on this approach, other papers were published to improve ELA. A key to success for this type of algorithms is to well distinguish the right interpolation direction which is normally done by modifying (2.19). Chen et al. in [28], defined two parameters

$$\begin{aligned} P &= \sum_{l=-\alpha+1}^{\alpha} \left|X(i-1,j-1) - X(i+1,j-l+1)\right| \\ Q &= \sum_{l=-\alpha+1}^{\alpha} \left|X(i-1,j-l+1) - X(i+1,j-1)\right| \end{aligned} \tag{2.20}$$

and modified (2.19) to

$$\hat{s} = \begin{cases} \arg\min_{-\alpha\leq s\leq 0} \left|X(i-1,j-s) - X(i+1,j+s)\right|, & P > Q \\ \arg\min_{-\alpha\leq s\leq \alpha} \left|X(i-1,j-s) - X(i+1,j+s)\right|, & P = Q \\ \arg\min_{0\leq s\leq \alpha} \left|X(i-1,j-s) - X(i+1,j+s)\right|. & P < Q \end{cases} \tag{2.21}$$

Wang et al [3] used the sum of absolute differences (SAD) to find the right direction as

$$SAD(m) = \sum_{i,j \in W} \left| X(i,j) - \frac{1}{2}(X(i-2m, j-2) + X(i+2m, j+2)) \right| \tag{2.22}$$

and

$$\hat{s} = \arg \min_{-2 \le m \le 2} SAD(m) \tag{2.23}$$

In [29] some predefined edge patterns are used to improve the performance of ELA.

In the rest of this section, we present our deinterlacing method based on FDI.

2.4.2 Fusion Based Deinterlacing Algorithm

As argued so far, the accuracy of edge-guided deinterlacing algorithms mainly depends on the discriminant applied to find the best interpolation direction. In the proposed algorithm, we first measure the error variances for each direction (σ_k), using our verification approach to find the two directions with the least errors and then fuse them under MMSE criteria

$$\begin{aligned} \hat{s}_1 &= \arg \min_{-2 \le k \le 2} \sigma_k \\ \hat{s}_2 &= \arg \min_{k \in \{-2,\cdots,2\} - \{\hat{s}_1\}} \sigma_k \end{aligned} \tag{2.24}$$

and

$$Y(i,j) = \frac{\sigma_{\hat{s}_2}^2 - \sigma_{\hat{s}_1,\hat{s}_2}}{\sigma_{\hat{s}_1}^2 + \sigma_{\hat{s}_1}^2 - 2\sigma_{\hat{s}_1,\hat{s}_2}} \hat{Y}_{\hat{s}_1}(i,j) + \frac{\sigma_{\hat{s}_1}^2 - \sigma_{\hat{s}_1,\hat{s}_2}}{\sigma_{\hat{s}_1}^2 + \sigma_{\hat{s}_1}^2 - 2\sigma_{\hat{s}_1,\hat{s}_2}} \hat{Y}_{\hat{s}_2}(i,j) \tag{2.25}$$

where $\hat{Y}_k(i,j)$ is the interpolated value of $Y(i,j)$ in the $(k+3)^{th}$ direction.

As depicted in Fig. 2.6, five directions are chosen to reconstruct missing lines in a field. For directions 2, 3 and 4 cubic convolution filter is used and simple averaging is used for directions 1 and 5.

2.5 Simulation Results

The proposed interpolation and deinterlacing methods based on FDI were implemented and tested on different scenes. The images shown below are the benchmarks used to test our interpolation technique. The images are selected among different scenes which are typically used in image processing applications. The original sizes are 512×512 and the LR images are obtained by direct down sampling by a factor of two (aliasing introduced).

First, the objective comparison between FDI and some of existing interpolation methods is presented. The objective metric applied is the PSNR. Table 2.1 shows the results of conducted simulation on images of Fig. 2.7

Table 2.1: PSNR (decibels) results of reconstructed images for methods bicubic, method in [2], method in [1], FDI and EC-FDI. For EC-FDI gains over the method in [1] are given in parentheses.

Image	Bicubic	method in [2]	Wu and Zhang [1]	FDI	EC-FDI
Cafe	21.39	21.32	21.56	**21.75**	**21.96**(0.4dB)
Baboon	22.91	22.82	22.88	**23.16**	**23.16**(0.34dB)
Flinstones	26.85	26.23	27.13	**27.66**	**28.00**(0.87dB)
Man	30.58	30.38	30.76	**31.05**	**31.22**(0.46dB)
Bush	27.61	26.92	27.20	**27.88**	**27.97**(0.77dB)
Lena	33.92	33.17	34.06	**34.70**	**34.90**(0.84dB)
Motor	38.30	35.80	38.39	**39.33**	**39.60**(1.21dB)
Fruits	36.97	36.10	37.16	**37.69**	**38.02**(0.86dB)
Average	29.81	29.17	29.89	**30.40**	**30.58**(0.72dB)

The gains over the method of [1] show that the modifications we introduced for EC-FDI in this study result in great objective improvements.

However the objective metrics only support an average measurement of perceptual quality. They can not take the effect of visual masking effect and other sharpening issues around edges into account [19]. Hence the subjective comparison is of more importance in image interpolation. Figs. 2.8 to 2.11 compare the results of FDI versus other methods for portraits of images: *Bush*, *Flinstones*, *Baboon* and *Cafe*.

The reconstructed images with bicubic interpolation suffer from blurred and jaggy artifacts around edges. The method in [2] reconstructs sharp edges, but the results are somewhat unnatural. This method is of the class of explicit edge-guided interpolation and error prone when the algorithm commits error in edge-detection. The method in [1] provides superior results than bicubic and [2], but still suffer from artifacts around edges and leaves discontinuities which are mostly removed by FDI. The results of EC-FDI are slightly superior than FDI regarding less noises and cleaner results. Error compensation does not affect the edge preserving issues which is addressed by directional filtering and optimal data fusion in this work.

The effect of considering additional horizontal and vertical interpolation directions to retain the sharp axial directions is depicted in Fig. 2.12.

The algorithm is also fast. The execution time for converting a 256×256, 8-bit grey-scale image to 512×512 is less than 0.8 second (Intel Pentium (4), 2.96 GHz, 1 GB RAM). Parallel processing is also applicable to speed up the algorithm since in each step, each missing pixel is computable individually.

The proposed de-interlacing method is compared with the method presented in [3] in Fig. 2.13. Our weighting method preserves the lines considerably better than the other method and results in sharper and cleaner edges.

2.6 Conclusion

Low complexity algorithms for 2D image interpolation and video de-interlacing are proposed. These methods are based on fusion of estimations in multiple directions. The proposed method of interpolation, FDI, in term of speed is comparable with fast algorithms e.g. cubic convolution, and is adaptive to edge orientations and varying pixel structures hence reduces the common artifacts and other visual defects of scene-independent algorithms which is verified via simulations.

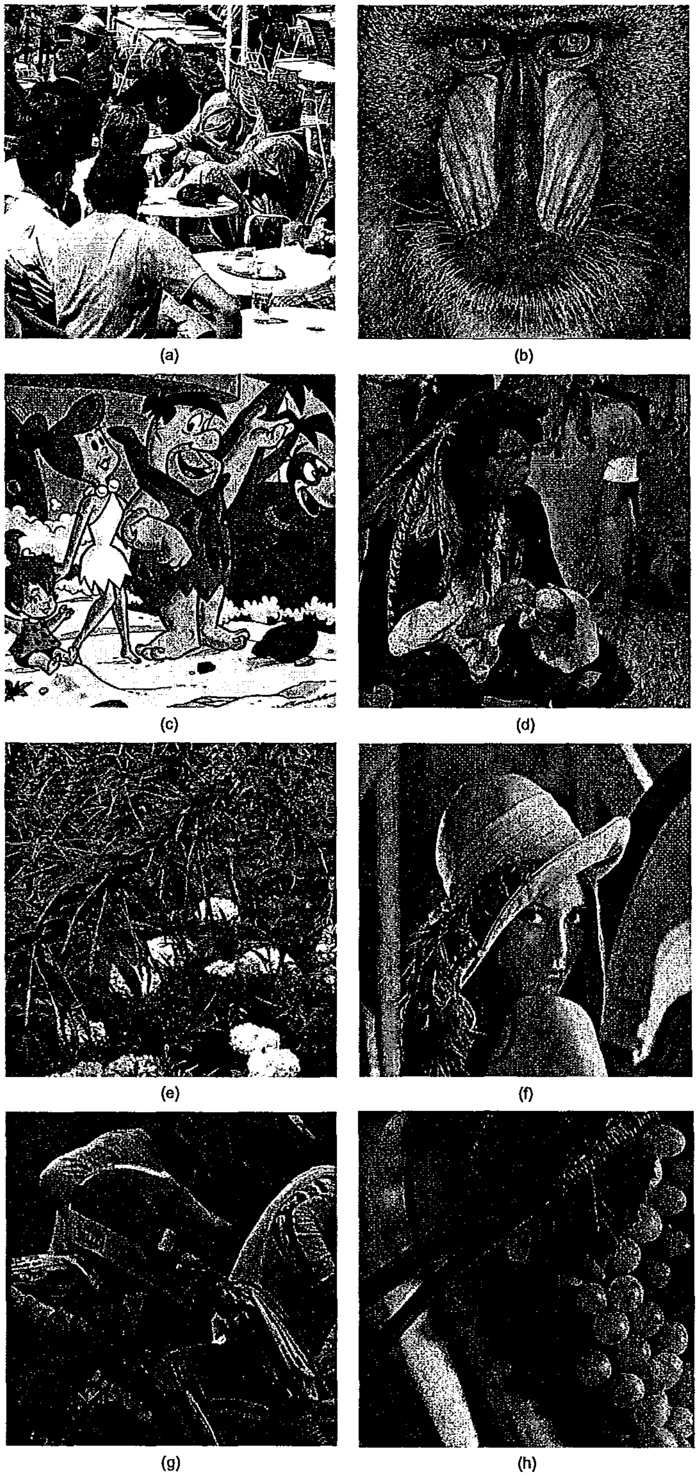


Figure 2.7: Sample images in the test set.

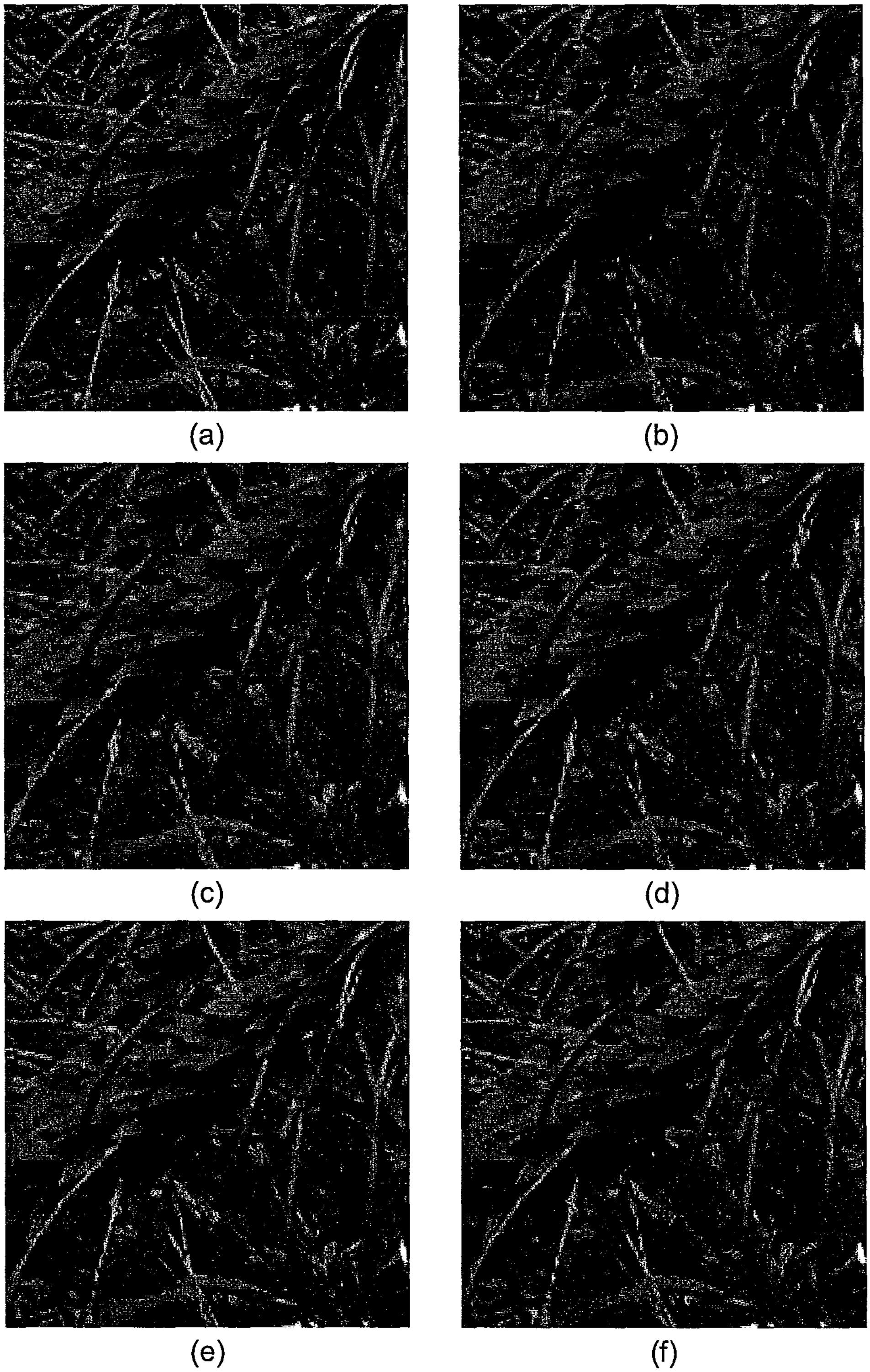


Figure 2.8: Portions of (a) original Bush image, (b) reconstructed image by bicubic , (c)method of [2], (d)method of [1], (e)FDI, (f)EC-FDI.

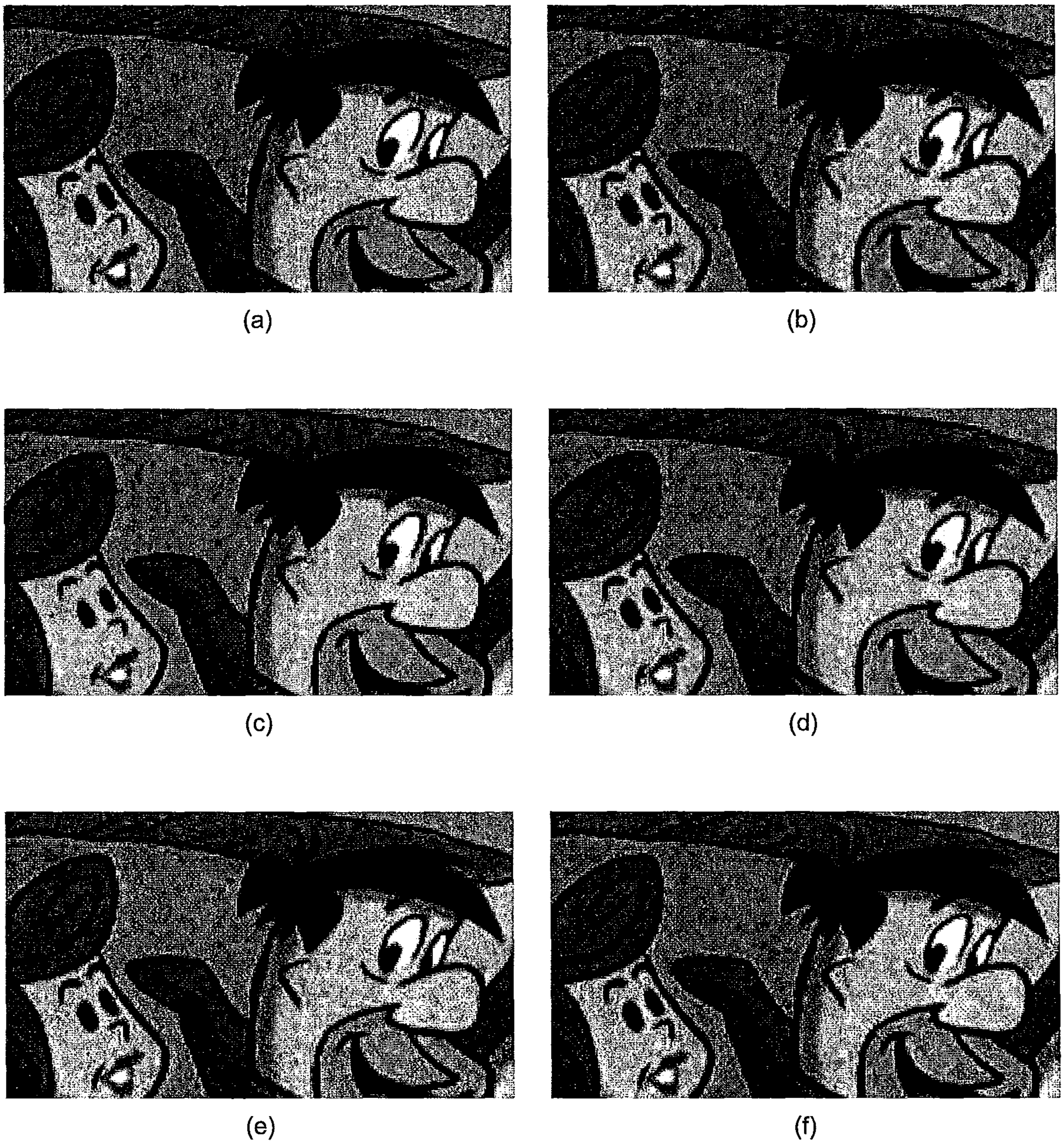


Figure 2.9: Portions of (a) original Flinstone image, (b) reconstructed image by bicubic , (c)method of [2], (d)method of [1], (e)FDI, (f)EC-FDI.

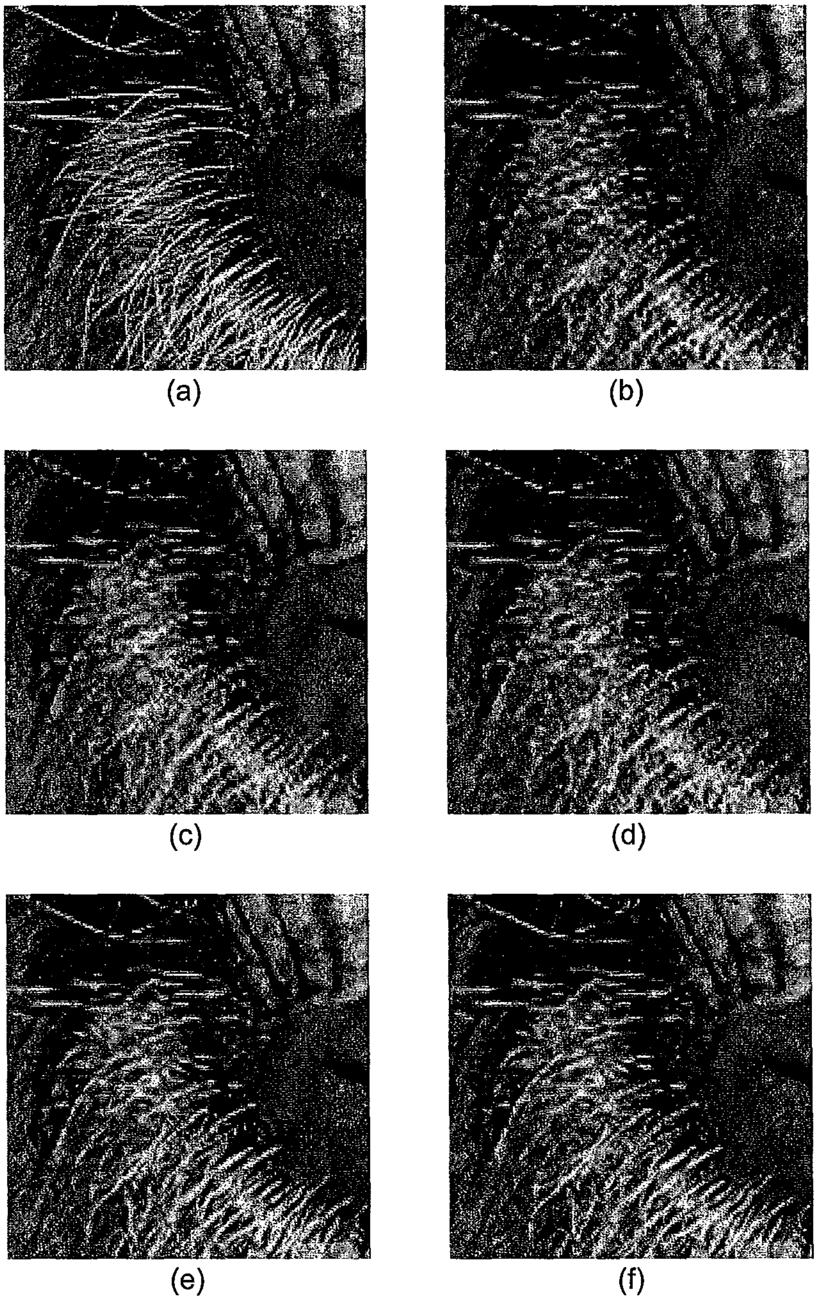


Figure 2.10: Portions of (a) original Baboon image, (b) reconstructed image by bicubic , (c)method of [2], (d)method of [1], (e)FDI, (f)EC-FDI.

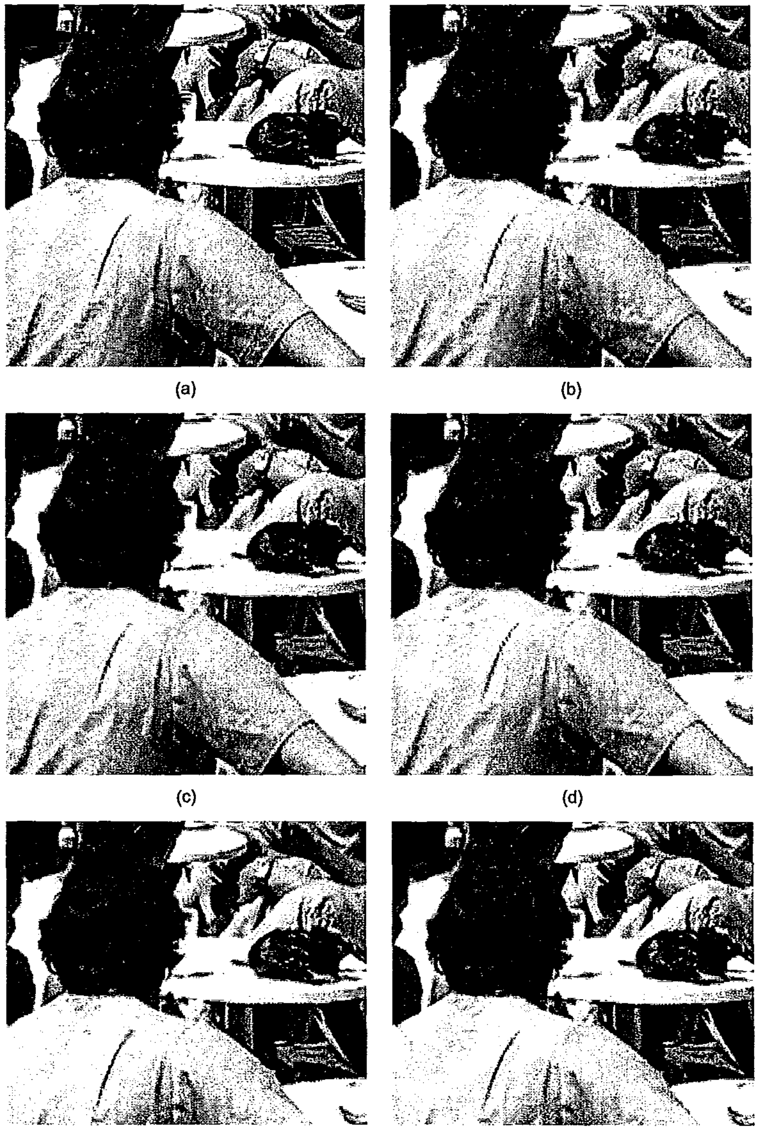
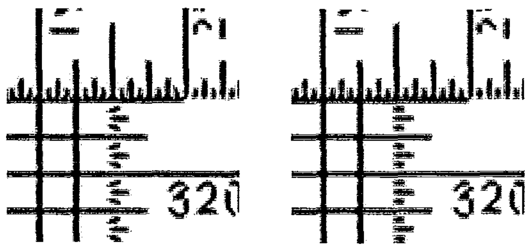
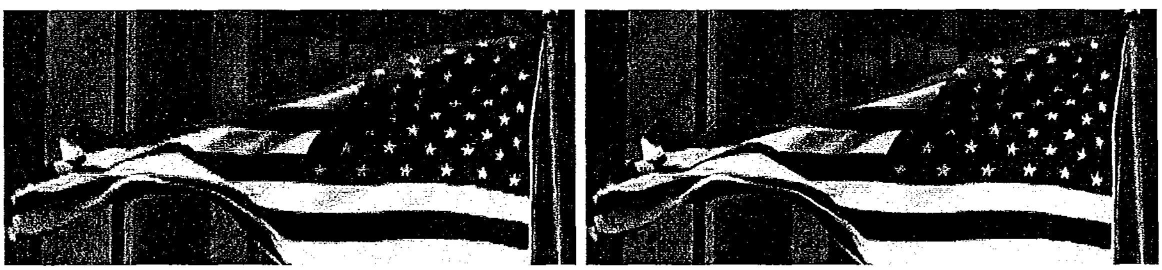


Figure 2.11: Portions of (a) original Cafe image, (b) reconstructed image by bicubic , (c)method of [2], (d)method of [1], (e)FDI, (f)EC-FDI.



(a) *12.85 dB* (b) *13.00 dB*

Figure 2.12: Portions of reconstructed image by (a)method of [1], (b)FDI.



(a) (b)

Figure 2.13: Results of deinterlacing by (a) method in [3], (b) proposed deinterlacing algorithm.

Chapter 3

Image Interpolation with Hidden Markov Model

For most applications the performance of an image resolution upconversion algorithm is judged by how well it can recover high frequency components of the underlying continuous light field. Different from the task of upsampling a one-dimensional signal, upsampling an image can and should exploit the fact that high frequency image features, such as edges and textures, are anisotropic in nature. A common technique adopted by many image interpolation algorithms is to estimate the edge/texture direction and interpolate in that direction [12], [15], [2] and [20]. However, there are two issues with these algorithms. First, the edge direction is estimated from the low-resolution input image, and it is error prone when insufficient sampling rate causes aliasing. Second, the edge direction is determined and directional interpolation carried out on a pixel by pixel basis disregarding the spatial coherence of edge points. In this work we address these issues and overcome the above said drawbacks by applying the hidden Markov model (HMM) technique to solve the problem of image resolution upconversion. The hidden states of the HMM correspond to edge directions or smooth waveforms of the underlying high-resolution image, which are not directly observable in the low-resolution image. Each of these HMM states is associated with an interpolator that suits the corresponding waveform. Ideally, if the HMM state was known for each pixel the corresponding interpolator would be applied for best performance. As such, the interpolation problem is converted to one of estimating the hidden states based on the observed low-resolution image, and it is solved by a maximum a posterior (MAP) sequence estimation method. As mentioned in Chapter 1, the advantages of our HMM-based interpolation approach are two folds: 1) it facilitates the incorporation of the statistics of high-resolution images of a training set into the interpolation process; 2) it makes a joint decision, via MAP sequence estimation, on interpolated pixels rather than on one pixel at a time in isolation.

The rest of this chapter is structured as follows. The general framework of interpolation with HMM is presented in Section. 3.1. The proposed edge-directed image interpolation and details of the algorithm are described in Section. 3.2. Simulation results and comparison between the proposed method and some existing methods are reported in Section. 3.3. Section. 3.4 concludes the chapter.

3.1 Hidden Markov Model for Image Interpolation

One dimensional hidden Markov model (HMM) theory was developed by Baum et al. in 1960s and is an effective machinery of characterizing sample dependencies in Markovian processes. It has had remarkable success in optimally estimating state sequences of Markovian processes given observed data, especially in speech recognition. Recently, some researchers proposed a two-dimensional extension of HMM

(2D-HMM), and applied the 2D-HMM to problems in image classification [30] and error-resilient image communication [31]. In this research we find the 2D-HMM to well suit the task of image interpolation, thanks to its ability to model two-dimensional spatial correlations.

The interpolation of a missing pixel can greatly benefit from the knowledge of waveform of the true underlying high-resolution image, in particular, whether the pixel is in a smooth area or near/on an edge. In the latter case it is crucial to know the edge direction. The difficulty is that the smoothness and orientation properties of the original image cannot be deduced reliably from the observed low-resolution image. This problem has a natural HMM formulation, if we classify the original 2D intensity function at the missing pixel $x_{i,j}$ into a set of states $S = \{\theta_0, \theta_1, \cdots, \theta_K\}$. Specifically in our application, θ_0 stands for the class of smooth waveforms in which an isotropic interpolator (e.g., bicubic) is effective, and θ_k, $1 \leq k \leq K$, for the class of edges of direction k. We quantize the edge directions into a small number K of classes to reduce algorithm complexity and also to avoid data overfitting in the construction of HMM using a training set. For direction k, a suitable directional interpolator will be used.

The interpolator cannot observe and thus needs to estimate the state $s_{i,j} \in S$ at pixel location (i, j) that is hidden by the down sampling process. The estimation is based on observable low-resolution features exhibited by $s_{i,j}$. In general, feature vector $\xi_{i,j}$ of $s_{i,j}$ consists of local attributes of the low-resolution image in a window $W_{i,j}$ centered at (i, j). In our setting, the interpolation of $x_{i,j}$ is solely determined by the state $s_{i,j}$.

In 2D-HMM the state $s_{i,j}$ of a pixel $x_{i,j}$ (or block of pixels) depends on the states of previous pixels (blocks) which are denoted by (Fig. 3.1)

$$\Omega_{i,j} = \{s_{\imath,\jmath} | (\imath \leq i) \cap (\jmath \leq j)\} - \{s_{i,j}\}. \tag{3.1}$$

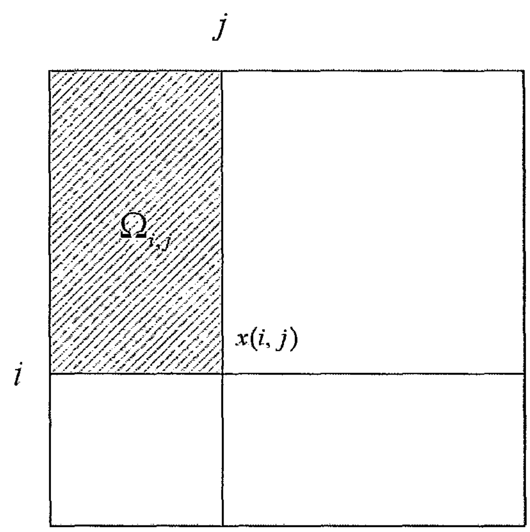


Figure 3.1: Definition of previous pixels in 2D HMM.

Assuming a first-order 2D-HMM, then the conditional probability of the hidden state $s_{i,j}$ of pixel $x_{i,j}$ is

$$P(s_{i,j}|\Omega_{i,j}) = P(s_{i,j}|s_{i,j-1}, s_{i-1,j}) \tag{3.2}$$

Using the HMM transition probabilities above, we formulate the interpolation of missing pixels on an $M \times N$ lattice $\mathbb{L}$ as a 2D MAP estimation problem of finding the state ensemble with maximum a posteriori probability [30]

$$s^* = \arg\max_s P(s|\boldsymbol{\xi}) = \arg\max_s \prod_{(i,j)\in \mathbb{L}} \Big\{ P(s_{i,j}|s_{i,j-1}, s_{i-1,j}) \cdot \tag{3.3}$$

$$p(\xi_{i,j}|s_{i,j}, s_{i,j-1}, s_{i-1,j}) \Big\}, \quad s \in S^{M\times N}$$

The above 2D-HMM estimation framework allows off-line learning to assist image interpolation. Indeed, the 2D transition probability $P(s_{i,j}|s_{i,j-1}, s_{i-1,j})$ can be learnt from an appropriate training set of high-resolution images whose statistics match those to be interpolated. The 2D-HMM also furnishes an adaptive image interpolator with an optimal way of exploiting 2D spatial correlations. But solving the 2D MAP problem (3.3) poses an algorithm challenge. Since a 2D image signal does not have a natural sequencing of the pixels, the classical dynamic programming algorithm for conventional 1D HMM problems cannot be directly applied. One possible approach is to lump all pixels of every row (or column) as into a "super-pixel" and consider the corresponding space of $(K+1)^N$ (or $(K+1)^M$) super HMM states. Then the 2D-HMM MAP estimation problem can be converted to one of 1D MAP sequence estimation by mapping the image to a sequence of super-pixels. This sequentialization would allow the use of dynamic programming algorithm. However, the above scheme is computationally intractable because the number of super states is of $O(K^{\min(M,N)})$, meaning this problem is NP-hard.

In quest for a practical algorithm for 2D HMM-based interpolation we propose to break down the 2D problem into two tightly coupled sequence estimation problems, each of which can be efficiently solved by dynamic programming. Granted such an approach can only produce an approximation solution of the 2D HMM MAP estimation problem, but we take careful considerations not to sacrifice the use of 2D spatial correlations in our algorithm design. In estimating the 2D state ensemble we make two orthogonal scans of the image: a horizontal scan followed by a vertical scan. When scanning row i, we fix the estimated states of row $i-1$ in (3.3) and compute the MAP state sequence for row i:

$$s_i^* = \arg\max_s \sum_j \Big\{ \log P(s_{i,j}|s_{i,j-1}, s_{i-1,j}^*) + \log p(\xi_{i,j}|s_{i,j}, s_{i,j-1}, s_{i-1,j}^*) \Big\}, s_i \in S^N$$

To develop a dynamic programming algorithm to compute the MAP state sequence s_i^*, we use the following recursion:

$$\begin{aligned} w_{i,k}(1) &= \log P(s_{i,1} = \theta_k) \\ &\quad + \log p(\xi_{i,1}|s_{i,1} = \theta_k, s_{i-1,1}^*), \quad 0 \le k \le K \\ w_{i,k}(n) &= \max_{1\le j\le K} \Big\{ w_{i,j}(n-1) \\ &\quad + \log P(s_{i,n} = \theta_k | s_{i,n-1} = \theta_j, s_{i-1,n}^*) \\ &\quad + \log p(\xi_{i,n}|s_{i,n} = \theta_k, s_{i,n-1} = \theta_j, s_{i-1,n}^*) \Big\} \\ &\quad 0 \le k \le K,\ 2 \le n \le N. \end{aligned} \tag{3.4}$$

By solving (3.4) for $n = 2, 3, \cdots, N$, we obtain

$$\max_s P(s, \boldsymbol{\xi}) = \max_{1\le k\le K} w_{i,k}(N), s \in S^N \tag{3.5}$$

and the resulting state sequence is s_i^*.

After estimating the corresponding state of any pixel (i, j), denoted by $s^h_{i,j}$, in the next step of underlying method, columns are scanned and the MAP sequence estimation is applied identical to previous step to estimate $s^v_{i,j}$. However, transition probabilities are updated based on estimates of row-scanning. Using Bayes' formula this can be formulated as

$$P(s_{i,j}|s_{i,j-1}, s_{i-1,j}, s^h_{i,j}) = \frac{P(s_{i,j-1}, s_{i-1,j}, s^h_{i,j}|s_{i,j})P(s_{i,j})}{P(s_{i,j-1}, s_{i-1,j}, s^h_{i,j})} \tag{3.6}$$

Assuming $s^h_{i,j}$ is independent of states of neighboring pixels in column-scanning, we have

$$P(s_{i,j-1}, s_{i-1,j}, s^h_{i,j}|s_{i,j}) = P(s_{i,j-1}, s_{i-1,j}|s_{i,j})P(s^h_{i,j}|s_{i,j}). \tag{3.7}$$

Using Bayes's formula

$$P(s^h_{i,j}|s_{i,j}) = \frac{P(s_{i,j}|s^h_{i,j})P(s^h_{i,j})}{P(s_{i,j})} \tag{3.8}$$

and

$$P(s_{i,j-1}, s_{i-1,j}|s_{i,j}) = \frac{P(s_{i,j}|s_{i,j-1}, s_{i-1,j})P(s_{i,j-1}, s_{i-1,j})}{P(s_{i,j})} \tag{3.9}$$

Plugging (3.7),(3.8) and (3.9) in (3.6) yields

$$P(s_{i,j}|s_{i,j-1}, s_{i-1,j}, s^h_{i,j}) = \frac{P(s_{i,j}|s^h_{i,j})P(s_{i,j}|s_{i,j-1}, s_{i-1,j})}{P(s_{i,j})} \tag{3.10}$$

With the assumption $P(s_{i,j}) \approx P(s^h_{i,j})$ the denominator of (3.10) is estimated from results of row-scanning by forward-backward algorithm [31]. Let in the estimated sequence s^*_i, the forward probability $\alpha_{i,j}(\theta_k)$ be the joint probability of being in state θ_k at $x_{i,j}$ and observing $\xi_{i,n}, 2 \leq n \leq j$. The backward probability $\beta_{i,j}(\theta_k)$ is defined as the conditional probability of observing $\xi_{i,n}, j < n \leq N$ given the state $s_{i,j}$ is θ_k.

As such

$$P(s_{i,j} = \theta_k) \approx \frac{\alpha_{i,j}(\theta_k)\beta_{i,j}(\theta_k)}{\sum_{\theta_i \in S} \alpha_{i,j}(\theta_i)\beta_{i,j}(\theta_i)}. \tag{3.11}$$

These probabilities can be computed by recursive formulae [30]

$$\begin{aligned}
\alpha_{i,1}(\theta_m) &= \pi_k p(\xi_{i,1}|s_{i,1}, s_{i-1,1}) \\
\alpha_{i,j}(\theta_m) &= p(\xi_{i,j}|s_{i,j}, s_{i,j-1}, s_{i-1,j}) \sum_{k=0}^{K} \alpha_{i,j-1}(\theta_k) P(s_{i,j}|s_{i,j-1}, s_{i-1,j}) \\
&0 \leq m \leq K
\end{aligned} \tag{3.12}$$

where π_k is the probability of being at k^{th} state at the first pixel of a sequence and

$$\begin{aligned}
\beta_{i,N} &= 1 \\
\beta_{i,j}(\theta_m) &= \sum_{k=1}^{M} P(s_{i,j}|s_{i,j-1}, s_{i-1,j}) p(\xi_{i,j}|s_{i,j}, s_{i,j-1}, s_{i-1,j}) \beta_{i,j+1}(\theta_m)
\end{aligned} \tag{3.13}$$

The conditional probability $P(s_{i,j}|s^h_{i,j})$ in (3.10) is estimated by training. Applying the transition probabilities (3.10) in the above described MAP sequence estimations, column-scanning is performed analogously.

So far we presented a general scheme for image interpolation based on HMM. Using this method, in the next section we will propose a practical explicit edge-guided image interpolation algorithm and describe it in detail.

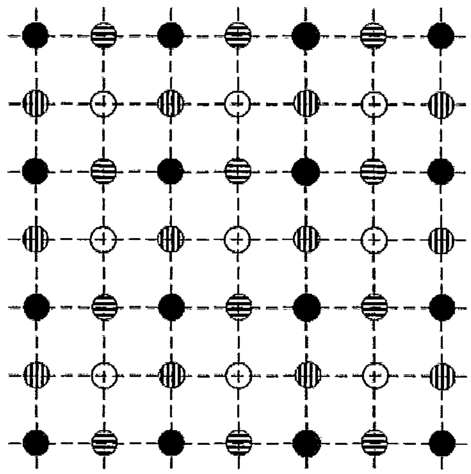


Figure 3.2: Formation of a high-resolution image form low-resolution image by up sampling. Solid dots represent original low-resolution samples. Empty and hatched circles represent missing pixels with different coordinates. Interpolation is done in two passes. First to find the empty circles, then to interpolate the hatched circles.

3.2 Edge Preserving Image Interpolation Based On HMM

The formation of a high-resolution (I_h) image from low-resolution (I_l) samples and estimated missing pixels is shown in the lattice depicted in Fig. 3.2. Three sublattices pictured by white ($\L_1$), vertically hatched ($\L_2$) and horizontally hatched ($\L_3$) circles are to be estimated. The 2D-HMM and MAP state estimation is applied on these sets of pixels separately to figure out the best interpolation direction for each interpolating pixel.

In the first pass of our algorithm the pixel sites in $\L_1$ with coordinates $I_h(2i, 2j)$ are interpolated. Directions specified in Figs. 3.3(a)-(c) are used in interpolating this set of pixels. These directions are of interest since in each there are four original samples from I_l in local window centered at $p_{2i,2j}$ ($W_{2i,2j}$) and hence are suitable for applying four tap filters e.g. cubic convolution. Nevertheless, in case of sharp edges in horizontal or vertical directions these diagonally aligned filters often suffer from large error where bicubic filter Fig. 3.3(c) outperforms its diagonally directed counterparts and is capable to retain these directions. Another substantial reason of applying bicubic as an 2-D isotropic filter is its well and sufficient performance in low frequency area where there is no dominant direction. In this case directional filtering magnifies some unreal lines and makes artifacts while bicubic filtering preserves smooth textures.

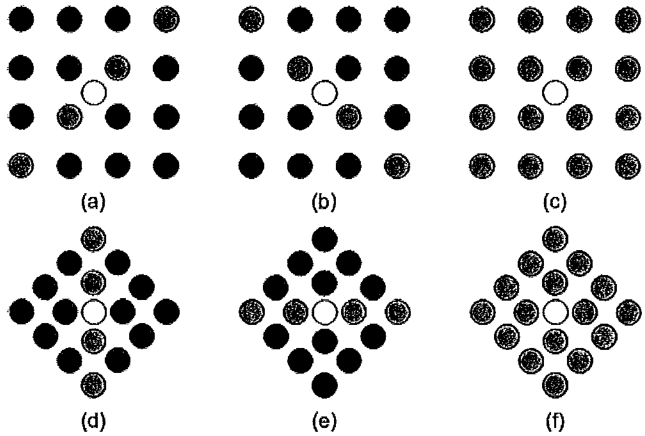


Figure 3.3: Sets of filters for proposed HMM-based image interpolation. (a)-(c) represent the directions to interpolate the pixels $I(2m, 2n)$ and (d)-(e) are associated directions for the pixel sites: $I(2m-1, 2n)$ and $I(2m, 2n-1)$.

Based upon the directions applied in the interpolation scheme, we need an observation providing us with information about texture orientation and frequency attributes of $W_{2i,2j}$. To address this problem we proposed a feature minimizing the linear least square problem below

$$\xi_{2i,2j} = \min_{\xi}^{-1} \sum_{m,n \in W_{2i,2j}} |I_l(m,n) - \xi \hat{I}_l^1(m,n) - (1-\xi)\hat{I}_l^2(m,n)|^2 \tag{3.14}$$

where $\hat{I}_l^1(m,n)$ and $\hat{I}_l^2(m,n)$ are interpolated values of known pixels by estimates of interpolation in directions depicted in Fig. 3.3. First unknown pixels are estimated in directions of (a) or (b) and then applied in a cubic filter to yield $\hat{I}_l^1(m,n)$ and $\hat{I}_l^2(m,n)$ respectively. The solution of (3.14), which is equal to (2.2), can be formulated as:

$$\xi_{2i,2j} = \frac{\sum_{m,n \in W_{2i,2j}} (\hat{I}_l^2(m,n) - \hat{I}_l^1(m,n))(\hat{I}_l^2(m,n) - I_l(m,n))}{\sum_{m,n \in W_{2i,2j}} |\hat{I}_l^1(m,n) - \hat{I}_l^2(m,n)|^2} \tag{3.15}$$

The Fig. 3.4 depicts six different portraits of variant textures and corresponding ξ for filters (a) and (b) in Fig. 3.3. Let direction 1 and 2 correspond to alignments shown in Fig. 3.3(a) and (b) respectively. The higher value of this feature in cases (c),(d), shows the prominent direction is along direction 1 and the edges in (a),(b) and (f) are close to direction 2. In presence of edges not along specified filters the value of ξ tends to $\frac{1}{2}$ e.g. Fig. 3.3(e).

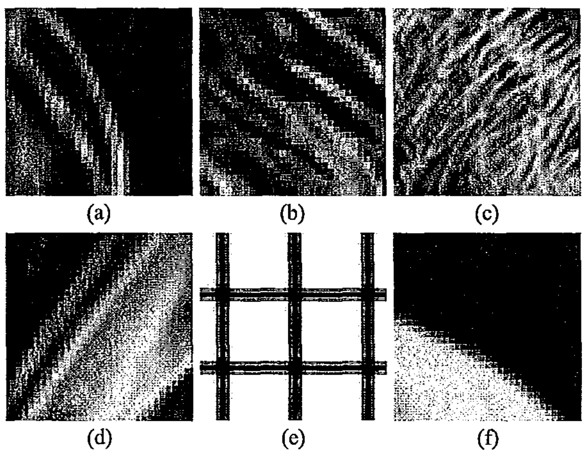


Figure 3.4: Portions of different structures. The average value of ξ in each picture is: (a) 0.76, (b) 0.89, (c) 0.24, (d) 0.35, (e) 0.49, (f) 0.77.

Based on this definition, we can solve the MAP estimation problem formulated in (3.4). The MAP estimation for every row/column is equivalent to finding the longest path in the graph shown in Fig. 3.5 The quantity of $w_i(N)$ is the overall weight for the longest path starting from s_1 to the end of sequence, which is computed by dynamic programming.

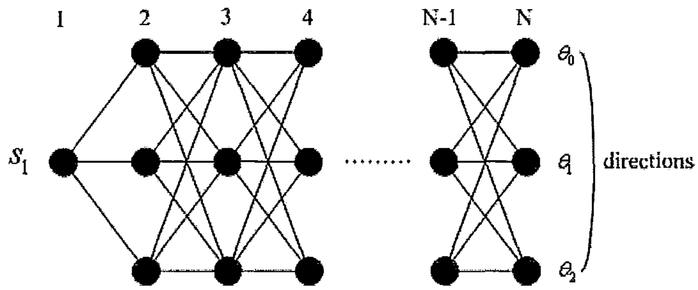


Figure 3.5: Graph constructed for MAP estimation.

To adapt the transition probability matrix by context of pixels we have classified contexts based on their activity. According to our classification a context is identified as a highly active context when the term $\sigma_1^2 + \sigma_2^2$ is above an empirically estimated threshold. In the low-frequency parts of an image both of these variances and hence their summation are relatively small. Based on contexts, two different transition probability matrices are used.

The observations are assumed to obey Gaussian distribution

$$p(\xi_{i,j}|s_{i,j}, s_{i,j-1}, s_{i-1,j}) = \frac{\exp\left\{-\frac{1}{2}(\xi_{i,j}-\mu)^T\Sigma^{-1}(\xi_{i,j}-\mu)\right\}}{(2\pi)^{\frac{J}{2}}|\Sigma|^{\frac{1}{2}}} \tag{3.16}$$

The quantities of mean vector (μ) and covariance matrix (Σ) are also estimated by training.

Up to now, we have described the interpolation process of the first pass. After estimating the samples in $\hat{I}_h(2i, 2j)$ of the high-resolution image, the remaining pixels of L_2 and L_3, $I_h(2i, 2j-1)$ and $I_h(2i-1, 2j)$ can be estimated in horizontal and vertical directions in a rotated $W_{i,j}$ by 45^o as shown in Fig. 3.3(d)-(f). Identical to the first pass, to deal with both sharp diagonal directions and low frequency parts of an image a 2-D filter with same coefficients as bicubic is used in a rotated window (Fig. 3.3(f)). The difference between first and second passes is unlike the first step, only in one of the axial directions shown in Fig. 3.3(d) and Fig. 3.3(e) original samples of I_l are available and the other direction consists of already approximated samples $\hat{I}_h(2i, 2j)$. Apparently, the estimates provided by original samples are more likely to be close to real missing values than those of preestimated ones. Thus the probabilities of transition to associated states of directions with original pixels are expected to be more, which is consistent with results of training.

This interpolation process is performed by row-scanning followed by scanning the columns. The resulting HR image is obtained by interpolating using filters corresponding to maximum a posteriori.

3.3 Simulation results

The proposed HMM-DI algorithm was implemented and tested on a variety of scenes.

Table 3.1: PSNR (decibels) results of reconstructed images for methods BICUBIC, NEDI and proposed HMM-DI.

Image	Bicubic	NEDI [19]	proposed HMM-DI
Lena	33.92	33.74	**34.56**
Flowers	36.79	36.50	**37.10**
Parrots	36.58	36.89	**37.25**
Motor	38.30	36.80	**39.20**
Flight	30.60	30.08	**30.95**
Peppers	31.84	29.73	**32.24**
Average	34.68	33.95	**35.21**

For objective comparison, the results of experiments on six test images listed in Fig. 3.6 are presented. The original images are 512×512 and the low resolution images are obtained by directly down sampling by a factor of two. These images were commonly used in previous works on image interpolation and for the fairness of comparison are selected among the cases with several edges making these images difficult for interpolation.

The PSNR results of applied methods on six sample images in Fig. 3.6 are tabulated in Table 3.1. The results are compared with the two of its predecessors: the conventional method of bicubic [6] and the edge-directed interpolation (NEDI), proposed by Li and Orchard in [19]. The first method is one of the computationally efficient but signal-independent filters and the second one is of the best structure adaptive methods with higher complexity. The objective comparison confirms proposed HMM-DI algorithm outperforms both competing methods, NEDI and bicubic. The average gain over NEDI is 1.26 dB. The bicubic is a proper benchmark for PSNR comparison, despite its moderate visual quality. The gain of 0.53 dB over bicubic interpolator verifies the good performance of HMM-DI in term of objective assessment. Especially, for the image Motor which contains several edges and rich frequency components the gain is around 1 dB over bicubic.

Objective measurements, e.g. PSNR, usually support a moderate evaluation of interpolation algorithms. For instance the sharpness and perceptual quality of recovered edges and other high frequency parts of an image cannot be directly assessed by the quantity of PSNR. Hence, in the remaining part of this section we conduct a subjective evaluation to assess the performance of proposed hidden Markov model based algorithm. The experimental results for two test images, Parrots and Flower, are depicted in Figs. 3.7 and 3.9.

As is evident in Figs. 3.6 to 3.9, the bicubic interpolator produces annoying jaggy artifacts along edges and provides inferior visual quality than NEDI and HMM-DI. As described before, applying an isotropic filter, with no spatial adaptivity to structures across an image is the main reason of producing jaggy edges and blurred details.

The edge-directed method of [19] is considered as one of the best interpolation algorithms so far and is very competitive in term of visual quality. This method is adaptive to local orientations and by estimating covariance of high-resolution image using low-resolution samples, is able to preserve the directions, especially long edge structures pleasantly. However it is subject to visual defects in highly active contexts which is visible in the curves around iris in Fig. 3.7 and the portraits shown in Fig. 3.10, where a side-by-side close up comparison between NEDI and proposed HMM-DI algorithm is provided. Since at any point the proposed method applies suitable filter among directional and isotropic alternatives it incorporates the advantages of a conservative method i.e. bicubic in low frequency areas and directional filtering around edges. Hence HMM-DI provides superior visual quality and preserves small curvatures better than NEDI.

3.4 Conclusion

An image interpolation algorithm based on hidden Markov modeling is proposed. The hidden states of the HMM correspond to edge directions or smooth waveform of the underlying high-resolution image. Each of these hidden states is associated with an interpolator that suits the corresponding waveform. The MAP sequence estimation is applied to find the most likely sequence of states for missing pixels in rows and columns. This solution is given by devised graph theoretical method. Simulation results verify the presented HMM based method outperforms competing methods both in term of objective measures (PSNR) and visual quality significantly.

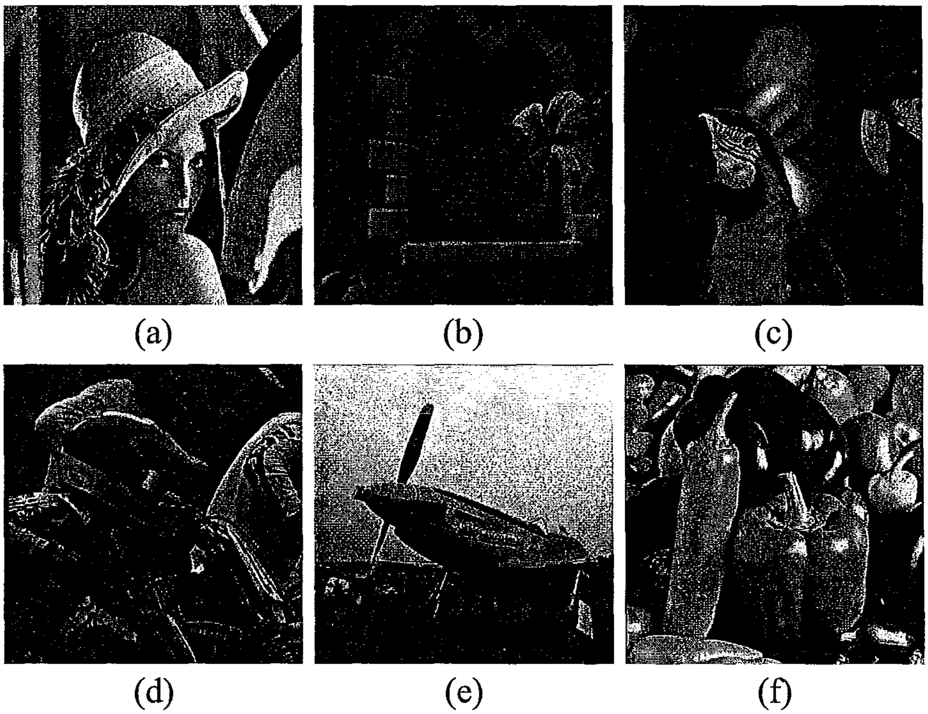
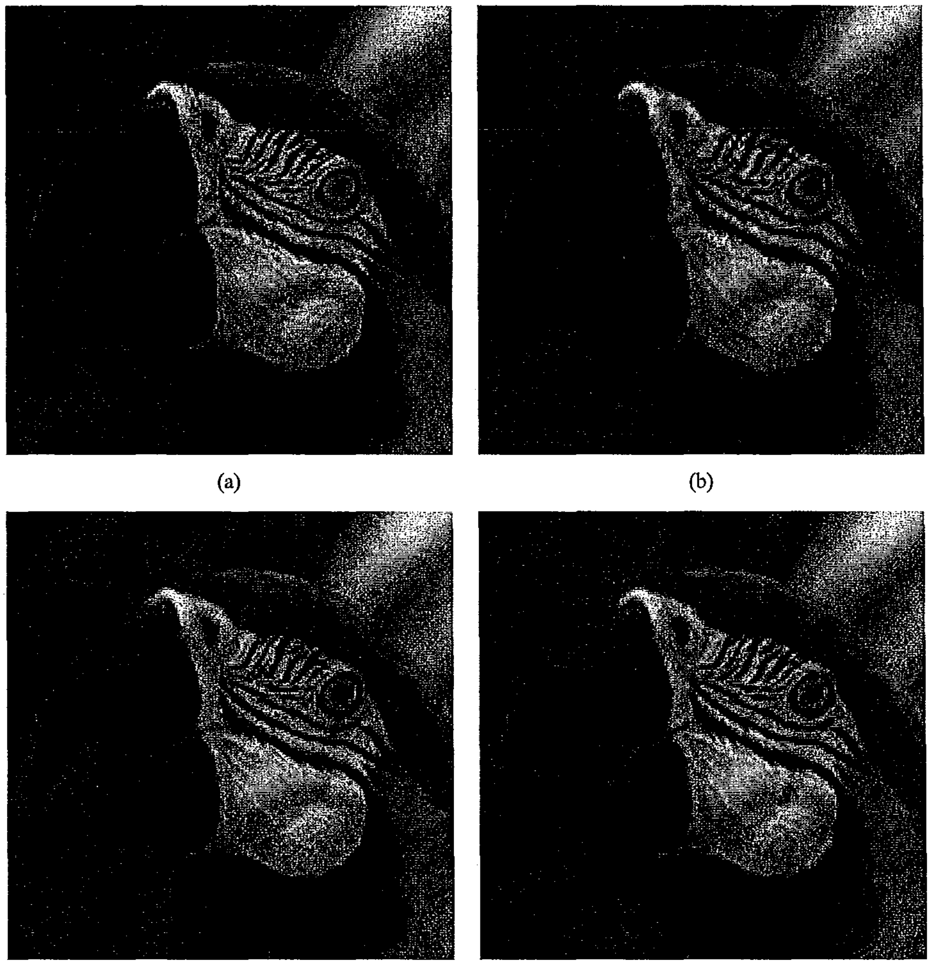


Figure 3.6: Sample images in the test set. (a) Lena, (b) Flower, (c) Parrots, (d) Motor, (e) Flight, (f) Peppers.



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

Figure 3.7: Comparison of different methods on a portion of the image Parrots. (a) original HRhigh-resolution image, (b) bicubic interpolation, (c) NEDI, (d) proposed HMM-DI.

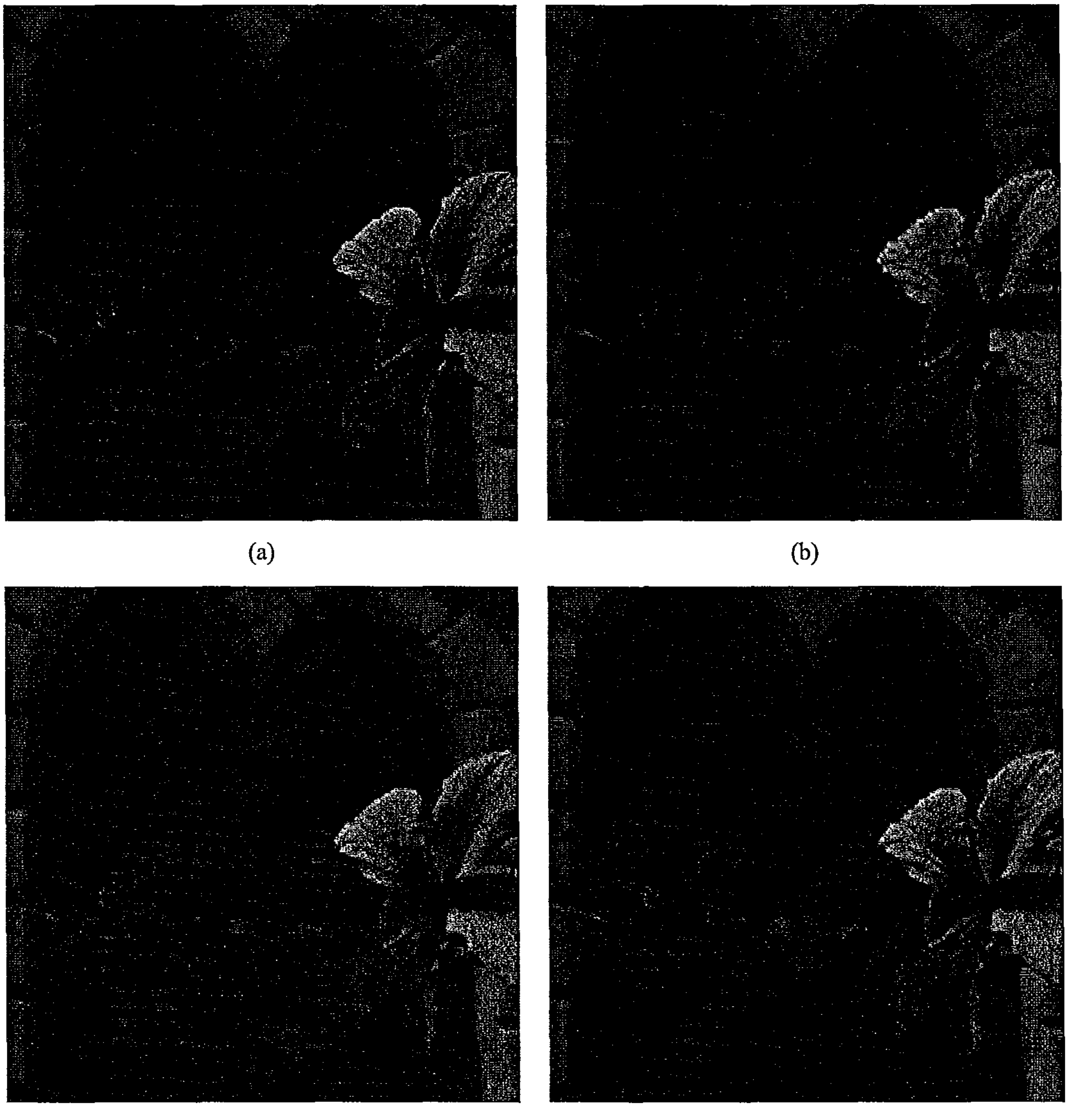

(a) (b) (c) (d)

Figure 3.8: Comparison of different methods on a portion of the image Flower. (a) original high-resolution image, (b) bicubic interpolation, (c) NEDI, (d) proposed HMM-DI.

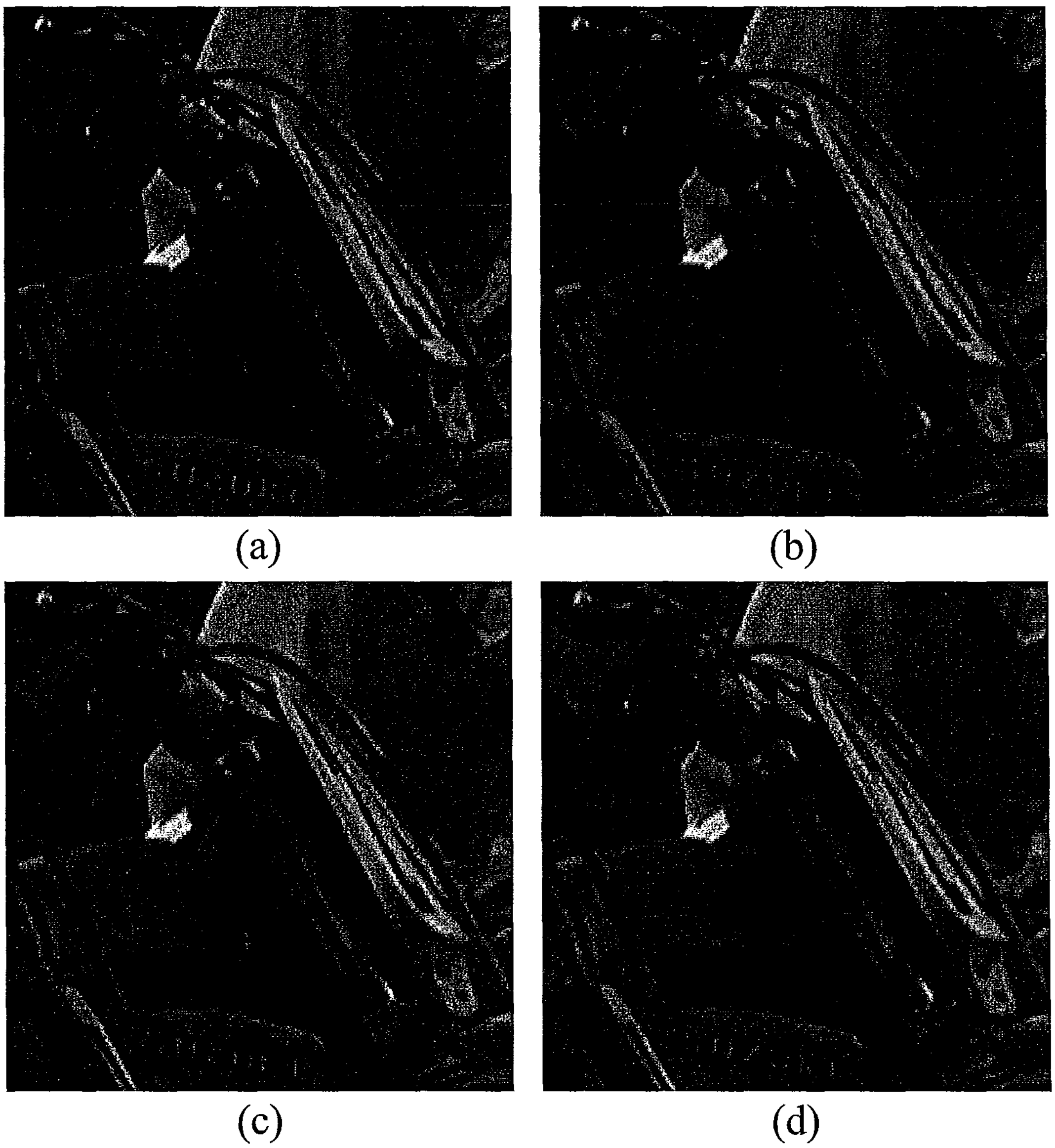


Figure 3.9: Comparison of different methods on a portion of the image motor. (a) original high-resolution image, (b) bicubic interpolation, (c) NEDI, (d) proposed HMM-DI.

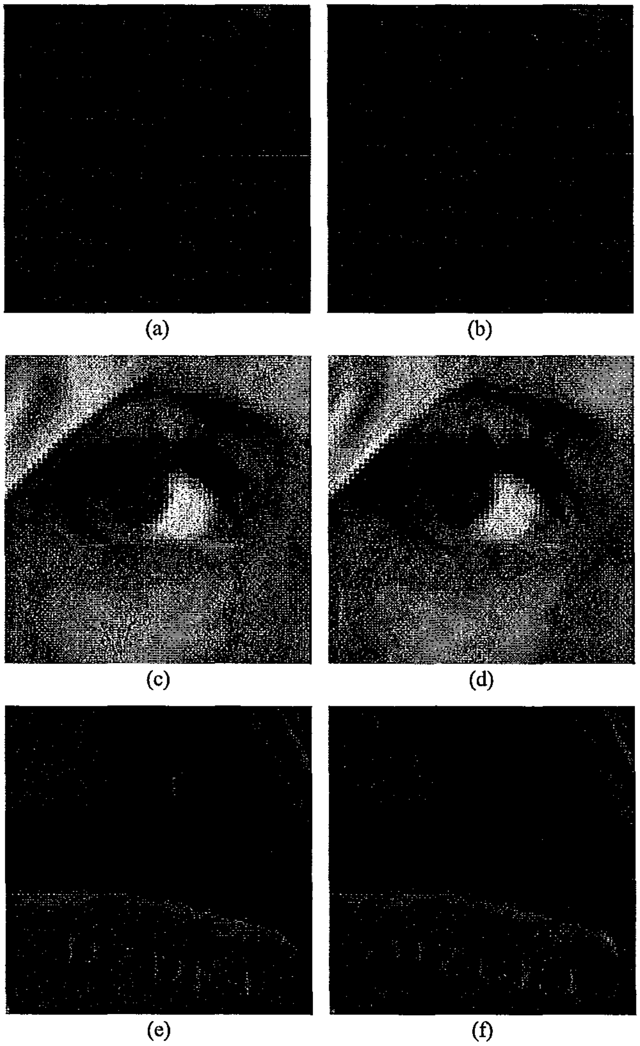


Figure 3.10: Close up visual comparison between NEDI and proposed HMM-DI. The portions on left column are reconstructed by NEDI and right column are interpolated by HMM-DI.

Chapter 4

Concluding Remarks

In this thesis we investigated the classical problem of image interpolation. Two different methods were developed: FDI and HMM-DI. Both techniques belong to the class of edge-directed image interpolation methods.

In FDI two estimates of a missing pixel by cubic interpolation in perpendicular directions are fused under the minimum mean square error criterion to improve the precision and robustness of the interpolation. In pursuit of better visual quality and cleaner high-resolution images, a context-based error compensation algorithm is also developed to reduce the error of cubic interpolator.

In HMM-DI the problem of image interpolation is converted to one of MAP sequence estimation by hidden Markov modeling. For each missing pixel the hidden state is associated with an interpolator that suits the corresponding waveform. HMM-DI incorporates the statistics of high-resolution images, which are supplied by a training set, into the interpolation process, and is able to exploit high-order dependency between missing pixels. Unlike the FDI (and most of existing interpolation techniques) where high-resolution pixels are interpolated one at a time independently, in

HMM-DI a sequence of missing pixels in row/column are jointly estimated.

Both of these algorithms are tested on a wide variety of scenes. These two new algorithms are compared with some of competitive existing methods in perceptual quality and objective quality (i.e., PSNR metric). The achieved reproduction qualities are competitive against the state of the art image interpolation techniques. Edges and fine details of the image are well preserved and interpolation artifacts plaguing many competing methods are greatly reduced.

4.1 Future Works

Even though the image interpolation methods proposed in this work show competitive performance compared to the best interpolation methods, there are some issues that call for further investigations in the future. Hereby, some of possible future works are outlined:

1. The idea of context-based error compensation is not restricted to cubic interpolator, but applicable for any linear interpolation filter. Furthermore, the selected features in the context classification are not chosen optimally. Therefore, the FDI algorithm can be improved in these two fronts.

2. The proposed optimal data fusion in Chapter 2 can be tailored to data of any dimension. It is possible to apply it in other applications, such as 3D medical image interpolation or video frame rate up-conversion.

3. In this thesis we introduced the idea of edge-guided image interpolation based on hidden Markov modeling. The problem formulation and the MAP estimation process are general. However, the observations associated with hidden states are

specific to this work. Finding better observation(s) to enhance the performance of the HMM-DI method can be another research topic.

Bibliography

[1] L. Zhang and X. Wu, "Image interpolation via directional filtering and data fusion," *IEEE Transactions on Image Processing*, pp. 2226–2238, August 2006.

[2] K. Jensen and D. Anastassiou, "Subpixel edge localization and the interpolation of still images," *IEEE Transactions on Image Processing*, pp. 285–295, March 1995.

[3] C. Wang, H. Han, and S. Peng, "A spatial deinterlacing algorithm based on edge orientation optimized in local area," pp. 87–90, in *Proc. congress on image and signal processing*, 2008.

[4] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing.* Prentice Hall, 1999.

[5] E. Meijering, "From ancient astronomy to modern signal and image processing," pp. 319–342, in *Proceedings of the IEEE*, 2002.

[6] R. G. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Transactions on Acoustic, Speech and Signal Processing*, pp. 1153–1160, 1981.

[7] C. G. T. M. Lehmann and K. Spizter, "Survey: interpolation methods in medical image processing," *IEEE Transactinos on Medical Imaging*, pp. 1049–1075, 1999.

[8] P. Thévenaz, T. Blu, and M. Unser, *Handbook of medical imaging.* 2000.

[9] L. Liang, "Image interpolation by blending kernels," *Signal Processing Letters, IEEE*, pp. 805–808, 2008.

[10] S. E. El-Kbamy, M. M. Hadhoud, M. L. Dessouky, B. M. Salam, and F. E. Abd-El-Samie, "A new edge preserving pixel-by-pixel (pbp) cubic image interpolation approach," pp. C11 1–9, *Radio science conference*, 2004.

[11] G. Ramponi, "Warped distance for space-variant linear image interpolation," *IEEE Trans. Image Processing*, pp. 629–639, May 1999.

[12] H. Shi and R. Ward, "Canny edge based image expansion," pp. I–785 – I–788., in *IEEE Int. Symposium on Circuits and Systems*, 2002.

[13] J. W. Hwang and H. S. Lee, "Adaptive image interpolation based on local gradient features," pp. 359–362, in *IEEE Signal Processing Letters*, March 2004.

[14] C. Vázquez, E. Dubois, and J. Konrad, "Reconstruction of nonuniformly sampled images in spline spaces," *IEEE Trans. Image Processing*, pp. 713–725, June 2005.

[15] J. Allebach and P. W. Wong, "Edge-directed interpolation," pp. 707–710, in *Proc. IEEE Int. Conf. Image Processing*, 1996.

[16] X. Wu and X. Zhang, "Image interpolation using texture orientation map and kernel fisher discriminant," pp. I–49–52, in *Proc. IEEE Int. Conf. Image Processing*, 2005.

[17] S. Carrato and L. Tenze, "A high quality 2× image interpolator," *in IEEE Signal Processing Letters*, pp. 132–135, June 2000.

[18] S. Dube and L. Hong, "An adaptive algorithm for image resolution enhancement," *in Singals, Systems and Computers*, pp. 1731–1734, 2000.

[19] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Transactions on Image Processing*, pp. 1521–1527, October 2001.

[20] M. Li and T. Q. Nguyen, "Markov random field model-based edge-directed image interpolation," *IEEE Trans. Image Processing*, pp. 1121–1128, July 2008.

[21] W. K. Carey, D. B. Chuang, and S. S. Hemami, "Regularity-preserving image interpolation," *IEEE Transactions on Image Processing*, pp. 1293–1297, September 1999.

[22] Y. Zhu, S. C. Schwartz, and M. T. Orchard, "Wavelet domain image interpolation via statistical estimation," pp. 840–843, in *Proceedings of International Conference on Image Processing*, September 2001.

[23] D. D. Muresan and T. W. Parks, "Prediction of image detail," pp. 323–326, in *Proceedings of International Conference on Image Processing*, September 2000.

[24] S. G. Chang and Z. Cvetković, "Locally adaptive wavelet-based image interpolation," *IEEE Trans. Image Processing*, pp. 1471–1485, June 2006.

[25] X. Zhang and X. Wu, "Image interpolation by adaptive 2-d autoregressive modeling and soft-decision estimation," *IEEE Transactions on Image Processing*, pp. 887–896, June 2008.

[26] K. A. Atkinson, *An Introduction to Numerical Analysis.* John Wiley and Sons, 1988.

[27] T. Doyle, "Interlaced to sequential conversion for edtv applications," pp. 412–430, in *Proceedings of* 2^{nd} *international workshop on signal processing of HDTV*, 1988.

[28] T. Chen, H. R. Wu, and Z. H. Yu, "Efficient deinterlacing algorithm using edge-based line average interpolation," *opt. Eng.*, 2000.

[29] M. K. Park, M. G. Kang, K. Nam, and S. G. Oh, "New edge dependent deinterlacing algorithm based on horizontal edge pattern," *IEEE transactions on consumer electronics.*, 2003.

[30] J. Li, A. Najmi, and R. M. Gray, "Image classification by a two-dimensional hidden markov model," *IEEE transactions on signal processing.*, February 2000.

[31] M. E. Sargin, K. R. A. Altinok, and B. S. Manjunath, "Conditional iterative decoding of two dimensional hidden markov model," pp. 87–90, in *Proc. International Conference on Image Processing*, 2008.