UNIVERSITY OF CALIFORNIA, SAN DIEGO

**Markov Random Field Edge-Centric Image/Video Processing**

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering
(Signal and Image Processing)

by

Min Li

Committee in charge:

Truong Q Nguyen, Chair
Serge Belongie
Pamela Cosman
William Hodgkiss
Nuno Vasconcelos

2007

The dissertation of Min Li is approved, and it is acceptable in quality and form for publication on microfilm:

---

---

---

---

---

Chair

University of California, San Diego

2007

In the memoriam of my second older sister

Hui-Ling Li (11/18/1973 - 2/03/1994)

*May you find comfort in the arms of the angel* $\cdots$

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

# LIST OF SYMBOLS AND ABBREVIATIONS

| | |
|---|---|
| MRF | Markov Random Field |
| GMRF | Gaussian MRF |
| GD | Gibbs' Distribution |
| 1-D, 2-D | One- or Two-Dimensional |
| $\mathbf{I}$ | An image; a field |
| $I(i,j)$ | Pixel value $I$ at pixel site $(i,j)$ |
| PIV | Pixel Intensity Variation |
| $Fi_o$ | Odd field |
| $Fi_e$ | Even field |
| $\hat{Fr}$ | De-interlaced frame from a field |
| $Fi_{refs}$ | reference fields in the temporal dimension |
| $\mathcal{N}$ | A general neighborhood structure |
| $\mathcal{N}_{i,j}$ | Neighborhood structure of pixel site $(i,j)$ |
| $\mathcal{C}$ | the clique set |
| $\mathcal{S}$ | the state space that consists of all possible configurations |
| $\Delta\mathbf{I}_k(i,j)$ | the PIV set of the $k$th digitized direction in the data window associated with pixel site $(i,j)$ |
| $U_s((i,j),I)$ | Single-pixel related energy of pixel site $(i,j)$ with pixel value $I$ |
| $\|\Delta I\|$ | The absolute value (magnitude) of PIV $\Delta I$ |
| $\mathcal{W}$ | Data window |
| $\mathbf{W}$ | Weighting vector |
| $\omega$ | A configuration of an MRF field |
| $U(\omega)$ | Energy function of $\omega$ |
| $V_c(\omega)$ | potential function associated with clique $c$ in $\omega$ |
| DAS | Discontinuity-Adaptive Smoothness |
| p.d.f. | probability density function |
| **F, L** | General random field or process |
| **f, l** | concrete realizations of a random field or process (1-D or 2-D). |
| LP | Line Process |
| LCD | Liquid Crystal Display |
| MAP | Maximum A Posteriori |

ACKNOWLEDGEMENTS

This thesis work is advised by my advisor, Prof. Truong Q Nguyen. To him, I owe my deepest gratitude. It has been a privilege for me to learn from him for five years. He gave me much freedom in doing research. He respects every idea from his students. I always feel encouraged after discussing research problems with him. He is very patient and supportive, especially when I was looking for a suitable topic to work on in the first couple of years of my PhD study. He not only directed my research work, but also gave me precious instructions and suggestions on my future career plan.

I would also like to thank my earlier teachers without whose enthusiasm and support I might never have come to this field for a PhD degree, in particular the advisor of my master degree, Dr. Chi-Wah KOK. To my earliest teachers of all, my parents, I owe the greatest thanks. They inspired my interests in learning.

I would also like to thank my thesis committee members, Dr. Pamela Cosman, Dr. Nuno Vasconcelos, Dr. Bill Hodgkiss and Dr. Serge Belongie for their invaluable feedback and constructive suggestions.

I am very grateful for the financial support from the Office of Naval Research, which is made available through my advisor, Professor Truong Q Nguyen.

I would like to thank my laboratory colleagues, Cheolhong An, Wei-Hsin Chang, Shay Har-noy, Ai-Mei Huang, Natan Jacobson, Meng-Ping Kao, Sanjeev Kumar, Yen-Lin Lee, Koohyar Minoo, Karl Ni, Ryan Prendergast, Carson Pun, Vikas Ramachandra, Dung Vo, Ottavio Campana, Mainak Biswas, Preethi Chandrasekhar and Gokce Dane for their help and friendship. Especially, I'll miss the small over-lunch talks that I had with my lunch partners, Cheolhong An and Dung Vo.

Last but not the least, my most profound thanks go to my husband and my loving family, which consists of my beloved parents, my oldest sister and her wonderful husband and daughter, and my younger brother and his lovely wife. I could not have done any of this without their unconditional support and spiritual companionship.

The text of Chapter 4 is adapted from parts of the materials that have been accepted for publication as: M. Li and T. Q Nguyen, "A De-Interlacing Algorithm

Using Markov Random Field Model," *IEEE Transactions on Image Processing* and as: M. Li and T. Q Nguyen, "Markov Random Field Model-Based Edge-Directed Image Interpolation," *IEEE Transactions on Image Processing*. The dissertation author was the primary researcher and the co-author T. Q Nguyen listed in this publication directed and supervised the research which forms the basis for this chapter.

The text of Chapter 5 is adapted from the material that has been accepted for publication as: M. Li and T. Q Nguyen, "A De-Interlacing Algorithm Using Markov Random Field Model," *IEEE Transactions on Image Processing* and also appears in part as: M. Li and T. Q Nguyen, "Discontinuity-adaptive de-interlacing scheme using Markov Random Field model," *IEEE International Conference on Image Processing* 2006. The dissertation author was the primary researcher of this publication, and the co-author listed directed and supervised the research which forms the basis for this chapter.

The text of Chapter 6 is adapted from the material that has been accepted for publication as: M. Li and T. Q Nguyen, "Markov Random Field Model-Based Edge-Directed Image Interpolation," *IEEE Transactions on Image Processing* and also appears in part as: M. Li and T. Q Nguyen, "Markov Random Field Model-Based Edge-Directed Image Interpolation," *IEEE International Conference on Image Processing* 2007. The dissertation author was the primary researcher of this publication, and the co-author listed directed and supervised the research which forms the basis for this chapter.

The text of Chapter 7 is adapted in part from the material that is published as: M. Li and T. Q Nguyen, "Optimal wavelet filter design in scalable video coding," *IEEE International Conference on Image Processing* 2005 and in part from the material that has been submitted for publication as M. Li, P. Chandrasekhar, G. Dane and T. Q Nguyen, "Low-Complexity and Low Bit-Rate Scalable Video Coding Scheme Using Inter-Layer Motion Vector Interpolation Techniques," *Asilomar Conference on Signals, Systems and Computers* 2007. The dissertation author was the primary researcher of these publications, and the co-author T. Q Nguyen listed directed and supervised the research which forms the basis for this chapter.

## VITA

| | |
|---|---|
| October 17, 1976 | Born, Hebei Province, China |
| 2000 | B. S., Tianjin University, Tianjin |
| 2000–2002 | Research assistant, Department of Electric and Computer Engineering, The Hong Kong University of Science & Technology, Hong Kong |
| 2002 | M. Phil., The Hong Kong University of Science & Technology, Hong Kong |
| 2002-2007 | Research assistant, Department of Electric and Computer Engineering, University of California San Diego |
| 2007 | Ph. D., University of California San Diego |

## PUBLICATIONS

**M. Li** and T. Q Nguyen, "Markov Random Field model-based edge-directed image interpolation," to appear in *IEEE Trans. on Image Processing*.

**M. Li** and T. Q Nguyen, "A de-interlacing algorithm using Markov Random Field model," to appear in *IEEE Trans. on Image Processing*.

**M. Li**, P. Chandrasekhar, G. Dane and T. Q Nguyen, "Low-complexity and low bit-rate scalable video coding scheme using inter-layer motion vector interpolation techniques," accepted by *Asilomar Conference on Signals, Systems and Computers* 2007.

**M. Li** and T. Q Nguyen, "Markov Random Field model-based edge-directed image interpolation," *ICIP'07*, September 2007.

**M. Li** and T. Q Nguyen, "Discontinuity-adaptive de-interlacing scheme using Markov Random Field model," *ICIP'06*, Oct. 2006.

**M. Li** and T. Q Nguyen, "Optimal wavelet filter design in scalable video coding," *ICIP'05*, vol.1, pp. 473-476, Sept. 2005.

**M. Li**, M. Biswas, S. Kumar and T. Q Nguyen, "DCT-based phase correlation motion estimation for video compression application," *ICIP'04*, vol.1, pp.445-448, Oct. 2004.

**M. Li** and C.-W. KOK, "Norm induced QMF banks design using LMI constraints," *ICASSP'03*, vol.6, pp.493-496, April 2003.

C.-W. KOK and **M. Li**, "Designing IIR filter bank composed of allpass sections," *ICASSP'03*, vol.6, pp.485-488, April 2003.

**M. Li** and C.-W. KOK, "Linear phase filter bank design using LMI-based $H_\infty$ optimization," *IEEE Trans. On Circuits and Systems II: Analog and Digital Signal Processing,* vol.50, no.3, pp.143-149, March 2003.

**M. Li** and C.-W. KOK, "Linear phase IIR filter bank design by LMI based $H_\infty$ optimization," *ISCAS'02*, vol.3, pp.437-440, May 2002.

ABSTRACT OF THE DISSERTATION

**Markov Random Field Edge-Centric Image/Video Processing**

by

Min Li
Doctor of Philosophy in Electrical Engineering
(Signal and Image Processing)

University of California San Diego, 2007

Professor Truong Q Nguyen, Chair

With the rapid development of digital video broadcasting technologies, the requirements on image qualities have been increased significantly. Furthermore, the computational power of today's processors is ever increasing and it becomes feasible to use more robust and complex algorithms to perform post-processing tasks without distorting edges seriously. In this dissertation, we explore the application of Markov Random Field (MRF) models in video post-processing. MRF models are able to model the spatial dependencies among pixels in an image and can impose spatial constraints on the processed images. They are a good fit for content-adaptive processing purposes.

We formulate the 2-D discontinuity-adaptive smoothness (DAS) constraint and impose it on the processed images via MRF modeling. This constraint assumes piecewise smoothness of images. However, the smoothness does not disturb discontinuity features, such as edges and object boundaries. It helps the processed image to achieve smoothness along edge directions and remain sharp in across-edge directions.

Local edge direction information is required when formulating the 2-D DAS constraint. Considering the limitations of the conventional binary representation, (edge direction, non-edge direction), regarding local edge directions, we propose a robust statistic-based approach to measure the local edge direction. In the pro-

posed method, local edge directions are represented using a weight vector. Using a rational number which is normalized to the range of zero to one, we provide the likelihood for each direction to be the true edge direction.

The formulated 2-D DAS constraint is applied to motion compensated (MC) de-interlacing and spatial interpolation problems. In MC de-interlacing, an efficient protection strategy against erroneous motion vectors is necessary. By the MRF model, the 2-D DAS constraint is imposed on the de-interlaced frame. The final output of each pixel site is the one that fits its context best under the 2-D DAS constraint. The context includes neighboring pixels from both the available field and newly interpolated field. Especially, the de-interlaced edges are expected to have strong geometric regularity. In spatial interpolation, we propose the MRF-EDI (Edge-Directed Interpolation) method, which uses the local edge direction information through the 2-D DAS constraint in an implicit manner to create clean and sharp edges.

Low-complexity implementations of the proposed algorithms are also investigated. One low-complexity implementation is to apply the proposed iterative optimization method only to near edge areas because simple, deterministic interpolation provides satisfactory results in smooth areas. Another low-complexity implementation is to replace the iterative optimization with a single-pass implementation. The complexity is reduced significantly by single-pass while the performance degradation is negligible.

In addition to video post-processing topics, this thesis contains some work on wavelet-based and H.264/AVC-based scalable video coding. In wavelet-based SVC, the Low Band Correction (LBC) technology is preferred for its efficiency in reducing overhead information and bit stream length. However, with the LBC technology, the lowpass filter implementing spatial scalability has to be a wavelet filter, which obeys the strong half-band condition. We develop a parametric design method to design a class of wavelet filters that considers all the design requirements and is suitable for both motion estimation and compression in SVC.

In the H.264 AVC-based scalable video model, motion vectors have to be coded and sent for all the spatial layers even in low complexity and low bit-rate applica-

tions, which is not efficient. We investigate the possibility to perform inter-layer motion vector prediction/interpolation in low bit-rate applications such that not all motion vectors have to be sent to the decoder side. Several motion vector prediction methods are combined and a novel mode-map is produced to indicate the chosen method for a block.

# 1

# Introduction

## 1.1 Development of the Display Technology and the New Digital Video Broadcast System

With the development and improvements of digitization and display technologies, traditional television broadcasting is undergoing profound changes. One change is the arrival of the digital TV era. The other one is the evolution of broadcasting. The delivery of television signals to homes has been evolving to the broadcasting of multimedia content to both homes and mobile devices [4]. The diagram in Fig. 1.1 shows the new broadcasting scenario. One can observe that, under the new mobile broadcast standards [5], the new digital video broadcasting (DVB) systems contain digital TV and HDTV broadcasting, broadcasting to battery-powered handheld receivers [6], and IP-based multi-media content and data delivery [7].

Interactive digital TV [8], for example, is representative of the new broadcasting project. It is already a reality in Europe. It enables interactive news, interactive sports, educational programming, home shopping, banking, video on demand, maps and driving directions, etc. It has the potential in Europe to replace computers as the primary e-commerce avenue.

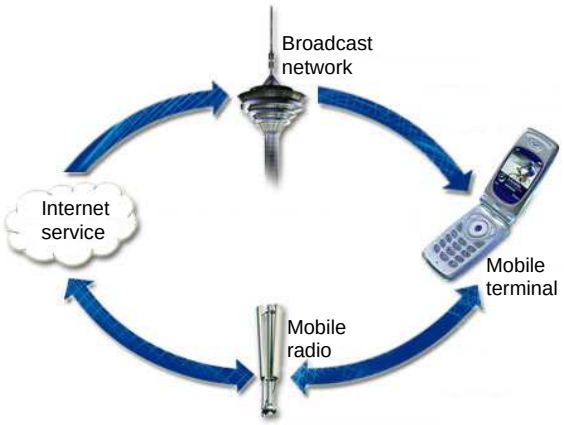



Figure 1.1: Typical mobile broadcast scenario (Duplicate of the diagram shown at http://www.dvb.com).

## 1.2 Post-Processing in DVB Systems

During the advancement from SDTV to HDTV, as well as from the traditional mobile phone to the mobile-TV enabled smart phone, the requirements on image qualities have been increased significantly [9]. Cinema-like picture quality is highly desired as the resolution of display devices increases to create more pleasant visual experiences. However, at high display resolution (the number of distinct pixels in each dimension that can be displayed), the noise which was "hidden" in the original picture and un-noticeable at low display resolutions becomes obvious and unacceptable [10]. Thus, high quality images to support high resolution displays are in demand, calling for efficient post-processing algorithms.

Video post-processing is the process of changing the perceived quality of a video on playback (done after the decoding process). Typically, there are three categories of post-processing algorithms in a DVB system. The first category aims to remove all kinds of distortion. The distortion is a result of lossy video compression that gives broken edges, smearing, ringing, and blocking artifacts. These artifacts make the visual quality of images unacceptable, especially on high resolution display devices. On the other hand, video compression is widely involved in DVB in

order to meet the transmission bandwidth limitation and is hard to avoid. For example, the baseband for terrestrial broadcasting of NTSC video is 4.2 MHz wide while the required bandwidth for full resolution HDTV is over 30 MHz. Efficient video compression technologies, for example, H.264, has to be used to reduce the bandwidth requirement.

In addition to distortion-removing post-processing, there are two other classes of post-processing. One is format conversion and the other is display-specific post-processing. Format conversion processing contains format changes in both temporal and spatial dimensions, i.e., frame rate changes in the temporal dimension and spatial resolution changes in the spatial dimension. The major format conversion in the temporal dimension is Frame Rate Up Conversion (FRUC). For example, if the received video sequence has a frame rate of 15 fps, by interpolating one frame between every two original frames, the frame rate can be converted to 30 fps. Major spatial format conversions include de-interlacing and scaling. De-interlacing is sampling rate doubling in the vertical dimension and scaling includes sampling rate conversion in both horizontal and vertical dimensions. The work in this thesis belongs to this category of DVB post-processing. More reviews and introductions of the previous work are available in Chapter 2.

The third category of post-processing algorithms is display-specific post-processing algorithms, which are proposed to enhance the displayed image quality based on the impulse response properties of LCD, such as error concealment, deblocking filtering, image sharpening and contrast enhancement.

## 1.3 Scalable Video Coding

Scalable Video Coding (SVC) is the video compression standard jointly developed by ITU-T and ISO. These two groups created the Joint Video Team (JVT) to develop the H.264 standard, or MPEG4 AVC video compression standard. In October 2003 the Moving Picture Experts Group (MPEG) issued a call for proposals on SVC Technology. Fourteen proposals, including twelve wavelet-based ones, were collected. The two remaining proposals were extensions of H.264/MPEG-4

AVC. One of them, proposed by the image team of the Heinrich-Hertz-Institute ([1]) (HHI) was chosen by MPEG as the starting point of its SVC standardization project.

The objective of SVC is to offer content in a "scalable" way. The video content is coded once. However, the produced single bit-stream can then be decoded at different bit-rates, requiring different computational power and channel bit-rate and offering streams of various spatial/temporal resolutions or qualities.

Video communications over the next-generation wireless networks are the main driving force of the development of SVC. This is because wireless channel conditions fluctuate in a wide range. Scalable video coding and adaptive services have been designed to adapt to the time-varying wireless environment. As wireless channel conditions change, the mobile terminal and network elements can scale the video streams and transport the scaled video streams to receivers at supportable bit-rates.

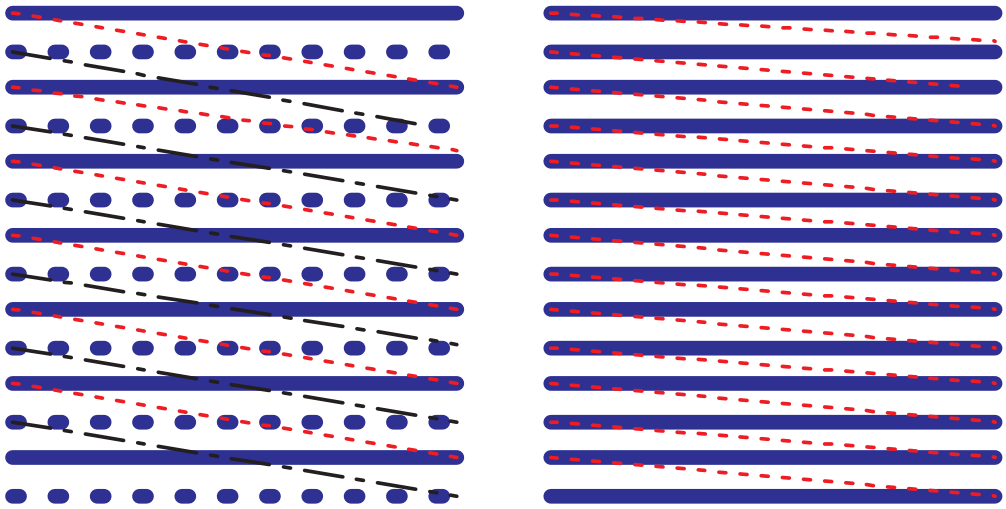
## 1.4 Focus of This Work

The work in this thesis focuses on

- de-interlacing
- spatial interpolation
- scalable video coding

The de-interlacing problem is introduced by two different scanning formats: *progressive* scanning format and *interlaced* scanning format as shown in Fig. 1.2. Progressive scanning scans a picture line by line while interlaced scanning scans every other line, where, if even-indexed lines are scanned at time $t-1$, odd-indexed lines will be scanned at time $t$. If a picture contains all the lines of a scene, it is called a *frame*. Otherwise, if it contains every other line in a scene, it is called a *field*.

The traditional analog TV broadcasting system adopted the interlaced scanning format as a standard in order to reduce large-scale flickering effects by taking



(a) Interlaced scanning (b) Progressive scanning

Figure 1.2: Progressive and interlaced scan formats.

advantage of the vision persistence effect of the human vision system (HVS) and achieve the tradeoff between data transmission bandwidth and vertical resolution of display. In the digital communication era, the interlaced scanning format is still widely used as an important strategy to achieve the tradeoff between large video data and costly communication bandwidth.

De-interlacing is the conversion from interlaced format to progressive format. People may ask why we need de-interlacing and whether the LCD display can support interlaced display. In principle, an LCD display can support interlaced display. However, the requirement that half of the pixels remain black in order to support interlaced display would result in up to 50% decrease of brightness. As a matter of fact, all current displays except for CRT screens require de-interlacing.

Spatial interpolation achieves spatial resolution conversion. To some extent, de-interlacing is a special kind of spatial/temporal interpolation, where only the resolution in the vertical direction is doubled. By spatial interpolation, we generally mean the resolution change in both horizontal and vertical dimensions. It's an important technique that supports the display of a single picture at various resolutions and is widely used in video communication over wireless networks. Normally, the bandwidth limitation of the wireless communication channel does

not allow transmitting video data at arbitrary frame rates and spatial resolutions. Consequently, pictures are transmitted at relatively low resolution, and at the receiver terminal, the decoded frames are then scaled to the desired spatial resolution via interpolations.

# 2

# Previous Work on De-Interlacing and Spatial Interpolation

## 2.1 De-Interlacing

From the signal processing point-of-view, de-interlacing is a vertical sampling rate up-conversion problem since the imaging process in a TV system is a spatio-temporal sampling process of the real scene [11]. According to the sampling theorem, a filter has to be used prior to sampling in order to remove frequencies that are beyond the Nyquist sampling rate so that all the original frequency components can be reconstructed by sampling rate up-conversion (de-interlacing). However, such a filter should be used in the optical path, which is not feasible for practical systems. As an ill-posed problem, it is impossible for the de-interlacing process to reconstruct all the frequency components (content) in the original frame. The de-interlacing process is the process of preserving the baseline spectrum and suppressing the aliasing artifact as best as possible [12].

If there is no motion, de-interlacing is not a difficult problem. As shown in Fig. 2.1, perfectly de-interlaced frames are obtained by weaving odd and even fields at adjacent time moments. If there is motion from time $t-1$ to $t$, directly weaving the neighboring fields will result in annoying artifacts as shown in Fig. 2.2. There are both camera and object motions in this example. The camera panning results

(a) Odd field at time $t-1$

(b) Even field at time $t$

(c) De-interlaced frame at time $t-1$.

(d) De-interlaced frame at time $t$.

Figure 2.1: An example of de-interlacing when there is no motion.

in global motion. In addition to the global motion, the bus is running from the right to the left and results in local motion.

Numerous algorithms have been proposed to solve the de-interlacing problem. More recent works can be found in [13–17]. According to [18], the available de-interlacing algorithms can be categorized into two classes, non motion compensated (Non-MC) de-interlacing and MC[1] de-interlacing. The following two subsections review these two classes of algorithms.

### 2.1.1 Non Motion Compensated De-Interlacing Algorithms

Non-MC de-interlacing algorithms include spatial/spatio-temporal de-interlacing [19–22] and motion adaptive de-interlacing algorithms [13, 14, 23, 24]. In [19] and [20], the de-interlacing is performed based on local edge direction information and the edge direction is determined based on the pixel intensity difference

[1]MC, depending on the context it is in, can be an abbreviation of motion compensation or motion compensated.

(a) Odd field at time $t-1$

(b) Even field at time $t$

(c) De-interlaced frame at time $t-1$.

(d) De-interlaced frame at time $t$.

Figure 2.2: De-interlacing artifacts when there is motion.

between neighboring pixel pairs in each discrete direction. However, with the presence of imaging noise, a single pixel intensity difference is not a reliable indicator of local edge directions. Besides, the considered discrete directions are only three or five, which are too few.

Methods proposed in [21] and [22] are median filtering de-interlacing methods. The final output of a site $s$ is expressed as $med$(A, B, C), where A represents the upper neighboring pixel of $s$, B represents the lower neighboring pixel of $s$ and C represents the temporal neighboring pixel at $s$. A more complex neighborhood structure can be used in the median filtering method.

The spatial and median filtering methods are preferred for their low implementing complexity but not performance. Commercial products of vertical linear de-interlacing and median filtering de-interlacing methods are available [25, 26].

The principles of motion adaptive algorithms are as follows. Motion detection between the reference field and the current field is first performed and the pixels to be interpolated in the current field are labeled as motion or non-motion pixels. Weaving of adjacent fields is used in the non-motion areas to increase the vertical resolution while, in the motion area, spatial de-interlacing is performed. The major difficulty with this class of methods is the motion detection part, where a decision of motion/no motion is made based on the magnitude of pixel differences. Due to camera motions, there is global motion from field to field and throughout a whole sequence. Consequently, motion detection indicates that there is motion everywhere. If that is the case, the motion adaptive interpolation results are not much different from those of spatial interpolation methods.

### 2.1.2 Motion Compensated De-Interlacing Algorithms

The non-MC algorithms perform well in stationary image regions, but are less effective with moving objects. This is because the Human Visual System (HVS) tends to favor motion-tracking. Serious flickering artifacts will be perceived if a moving object has not been interpolated consistently from frame to frame.

The MC de-interlacing algorithms [15–17] are advanced and perform better with moving objects, given accurate motion information. However, the direct

MC de-interlacing method, in which the de-interlacing processing is performed according to motion vectors (MVs), often does not produce satisfactory results for the overall video sequence. This is because MVs, which are usually generated from oversimplified motion models, can not describe all the temporal changes, such as occlusion, concealment, fades and scene changes. Consequently, an efficient MC de-interlacer must include effective protection strategies, which can protect the de-interlacer's performance against incorrect MVs. Failure near edge or boundary areas results in unpleasant artifacts since edges and boundaries in an image are more noticeable to the HVS. On the other hand, visual quality improvements in near edge or boundary areas significantly improve the overall visual quality.

Motion estimation (ME) and protection against incorrect MVs are two critical factors for a successful MC de-interlacing algorithm. We discuss field ME in subsection 2.1.3 and protection strategies in subsection 2.1.4.

### 2.1.3 Motion Estimation in MC De-Interlacing Algorithms

Unlike ME in coding, where the purpose is to reduce the energy of residual error images as much as possible, ME for de-interlacing aims to find "true" motion and the resulting motion vector field (MVF) does not necessarily lead to the minimal energy level of residual error images.

In terms of "true" ME, two models are promising. One is the 3-D Recursive Search (3DRS) [27–29] model and the other is the Global ME (GME) model [30]. The 3DRS ME model [29] estimates translational motion only. Its major difference from the common full search method is that the spatial and temporal continuity constraint of motion is imposed on the estimated MVFs. It assumes that the motion in the current block is the same as at least one of its neighboring blocks. Thus, the produced MVF is continuous temporally or spatially in at least one direction. This ME model fails where the motion continuity assumption is not true, for example, motion of small objects.

The GME model in [30] uses an affine model to estimate translational, rotational and zooming motions simultaneously. This model is efficient in estimating background (camera) motion. Motion of blocks that have different motions

from the background is estimated by local block-matching search. A bi-directional search can help occluded and newly appearing blocks. Both translational and affine models are oversimplified versions of the real projecting imaging process. Incorrect MVs are often produced in practical implementations.

The model-related difficulty is true for both field and frame MEs. In addition, there is structural difficulty with field ME. When a field is used as the reference image for MC, MVs frequently lead to unavailable pixels. Consequently, simple interpolation methods are used to generate these pixels. In this case, even when the motion information is correct, it is hard for the motion compensated candidate, which has been interpolated using a simple method in another context, to be better than the spatial interpolation candidates, which have been interpolated in the to-be-interpolated pixel's perceiving context. Because of the structural difficulty, there is lack of agreement on the optimal field ME structure. The three commonly used ones are reviewed below. In the following discussion, Frame MC means the MC process with the frame being the reference image and Field MC means the MC process with the field being the reference image.

**Frame-Field ME and Frame MC**

In this structure, the ME and MC are between the current field and previously reconstructed frame. No interpolation is necessary in this structure since the reference image is a frame. If the MVs are integers, for any pixel site in the current frame, the motion vector can locate a MC candidate in the reference frame.

The major disadvantage of this structure is that, if the reference frame has not been de-interlaced properly, the reliability of the resulting MVF will decrease. Consequently, de-interlacing artifacts will propagate temporally. In addition, the de-interlacing of the current field can not start until the the previous field has been de-interlaced. This may cause unnecessary delay.

**4-Field ME and Field MC**

In this structure, suppose the current field corresponds to time $t$. We denote the field at time $t-1$ as the forward field, the field at time $t-2$ as the forward-

forward field, and the field at time $t+1$ as the backward field. Chang, *et al.*, in [31] suggests the following field ME structure. Combine the forward-forward field with the forward field to obtain the reference frame and combine the current field with the backward field as the current frame. The ME is performed between the reference frame and the current frame. The obtained MVs are then scaled by a factor of 2 for MC, which uses the forward field as the reference image. Interpolation is involved in the MC process since the reference image for MC is a field.

This ME structure assumes that the translation-only motion from time $t-2$ to $t+1$ does not change direction or speed. With this assumption, motion from the current field to the forward-forward field is the same as that from the backward field to the forward field. By combining two fields, the ME result is more reliable than the result of ME between two fields only. Incorrect MVs will be produced when the assumption for motion from $t-2$ to $t+1$ is not true, as in the case of head-turning motion.

**Same Parity Field ME and Field MC**

This structure is used when implementing GME model. The global motion between the current field and the forward-forward field is estimated and the forward-forward field is used for global MC. For blocks that move differently from the background, bi-directional local block-matching search is used, with the forward-forward field and the backward field being the forward and backward reference fields respectively. Bi-directional ME benefits the occlusion or new appearing blocks. Simple interpolation is involved in ME between the current field and the backward field since they have opposite parities.

### 2.1.4 Protection Strategies Against Erroneous MVs

Protection strategies or robustness of incorrect motion vectors are essential for a successful MC de-interlacer. Frequently used protection strategies are the threshold value method [31], (weighted) median operations [32, 33], and the Adaptive Recursive (AR) method [34].

#### Threshold Value Protection

The threshold value method [31] decides the reliability of MVs based on the associated SAD (Sum of Absolute Difference) or MSE (Mean Square Error) levels in ME.

#### (Weighted) Median Filtering Protection

In median filtering protection, the median value of a pixel's MC candidate, vertically upper neighbor and lower neighbor is used as the final output. The deficiency of this method is that the motion compensated candidate is often filtered out and the edges get blurred.

#### Adaptive Recursive Protection

AR protection produces the output as a linear combination of the MC candidate and the spatial interpolation candidate (linear averaging interpolation in the vertical direction), for example, $pC_m + (1-p)C_v$. The linear combination factor $p$ is used to decide how closely the two vertical neighboring pixels are related to their MC candidates. This method can automatically switch the final output between MC candidate $C_m$ and vertically interpolated candidate $C_v$ to yield the final output. Error propagation in temporal dimension is a major disadvantage of this method.

## 2.2 Spatial Interpolation

Image interpolation is the process of producing a high resolution image from its low resolution counterpart.

### 2.2.1 Polynomial-Based Interpolation

The conventional image interpolation methods are bilinear, bicubic and spline [35] methods. This class of methods assumes continuity of the pixel intensity field, which is unrealistic. Continuity near edges only exists along edge directions while

abrupt discontinuities might happen across edge directions. Typically, images with blurred edges are produced in these methods. They are preferred not for their performance, but for the low computational complexity. There are several methods to improve these algorithms, i.e., [36]. [36] models pixel relations as fuzzy inferences. Consequently, the slope and the shape of the linking curve from one sample to another is controlled by a contrast enhancement function.

### 2.2.2 Improved Edge Interpolation

To improve the interpolation performance with edges, one class of methods is edge-enhancement post-processing and the other class of methods is spatial adaptive interpolation [37] [38] [39]. Edge-enhancement post-processing performs edge detection or high-pass filtering of the interpolated image followed by edge sharpening operations, such as sharpening mapping curves [40] and inverse heat diffusion equations [41]. The major difficulty of this class of methods is with the explicit edge detector, which has difficulties in detecting natural edges. More discussion can be found below. High-pass filtering methods will sharpen edges and amplify noises as well.

Spatial adaptive interpolation methods are another class of methods to improve edge interpolation. In this class of methods, the interpolation coefficients are adjusted according to the local pixel intensity properties [42, 43]. Interpolation along ideal step edges is not difficult since accurate edge direction information can be obtained explicitly from edge detectors [44–47]. However, edges in natural images appear as spatially blurred edges due to sensor noise, focal blur, penumbral blur, and shading, etc [48]. When edges are blurred or noisy, it is difficult to explicitly specify their characteristics, which makes detection of natural edges difficult.

### 2.2.3 Implicit Edge-Directed Interpolation

To avoid the difficulties with explicit edge-directed interpolation methods, implicit edge-directed interpolation methods are proposed. In this class of methods, the edge directions are not explicitly extracted. However, the interpolation is per-

formed based on implicit edge direction information.

New Edge-Directed Interpolation (NEDI) [49] is an implicit edge-directed interpolation method. In this method, the to-be-interpolated pixel's twelve causal neighboring pixels are vectorized and so are the pixel's four diagonal neighboring pixels. All the vectorized pixels are regarded as short-time realizations of a one-dimensional (1-D) stationary Gaussian process. The to-be-interpolated pixel is predicted from the short-time realizations of this process using Wiener filtering theory. The covariance matrix of the short-time realizations is required for the prediction. The so-called high resolution covariance matrix is interpolated from its low resolution counterpart according to geometric duality principles. The edge direction information is included in the low resolution covariance matrix and is preserved during the interpolation.

# 3

# Markov Random Fields

In the 1920's, Markov Random Fields were presented as a new type of stochastic process in probability theory. They had been broadly used in solving various problems, including statistical mechanics where it originated. Their first usage in image processing can be tracked back to 1972 [50] by Besag in processing binary data. Its applications in image processing became popular in 1980s with the publication of Geman & Geman's paper [2].

## 3.1 About MRF

### 3.1.1 Origin of MRF

Markov random fields originate from the *Ising* model in statistical physics. Consider the following 1-D Ising model example. As shown in Fig. 3.1, there can be several sites on a line. At each site, there is a small spin. Given any moment, the spin is in one of two positions, "up" or "down" and is described by function

$$\delta_i(\omega) = \begin{cases} 1 & \text{if } \omega_i \text{ is up} \\ -1 & \text{if } \omega_i \text{ is down} \end{cases} \tag{3.1}$$

The total magnetization with all the spins is a random variable and can be calculated according to

$$U(\omega) = -J \sum_{i,j} \delta_i(\omega)\delta_j(\omega) - H \sum_i \delta_i(\omega), \tag{3.2}$$

where $J$ is a material parameter ($J > 0$ corresponds to attractive interaction while $J < 0$ corresponds to repulsive interaction) and $H$ is the external magnetic field intensity.

To simplify the modeling task, only neighboring spins are assumed to have interactions. If two spins are not neighbors, their interaction is too weak to model. The probability measure is then defined by the Gibbs distribution,

$$P(\omega) = \frac{\exp\{-u(\omega/T)\}}{Z}, \tag{3.3}$$

where $Z$ is a normalizing parameter and the original meaning of $T$ is temperature, which controls the "annealing" speed to the stable state. $T$ becomes a global convergent control factor in image/video processing problems [51]. At sufficiently low temperature, even if the spins were random to begin with, they would tend to converge to a lower energy state. If there is no exterior field, this is completed by interactions between spins.

In this model, the desired appearance as a whole is achieved by the interactions between spins. This relationship can be interpreted in terms of images. Desired image properties such as smoothness and consistency can be achieved by defining the interactions among pixels.

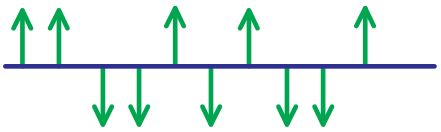
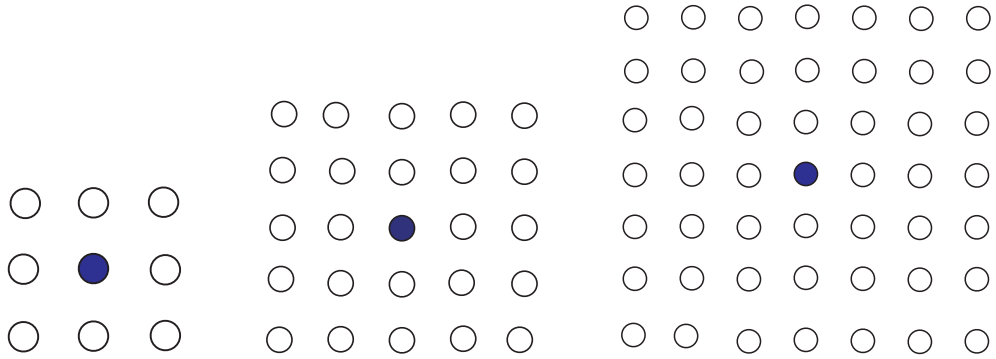


Figure 3.1: Magnetization of spins.

### 3.1.2 Basic Concepts and Theories about MRFs

#### Neighborhood Structure and Cliques

To define a MRF, a *neighborhood structure* $\mathcal{N}$ is required. The neighborhood structure defines the range of immediate interactions between one pixel and other pixels. The contribution of this single pixel to the whole image is made through



(a) 8-pixel NS (b) 24-pixel NS (c) 48-pixel NS

Figure 3.2: Three different neighborhood structures (NS). Non-filled circles represent neighboring pixels.

the immediate interactions to its neighboring pixels. Fig. 3.2 shows three different neighborhood structures. Circles represent pixels. The blank circles are neighbors of the filled circles. Neighborhood relationships are symmetric. If pixel $I(i, j)$ is one of pixel $I(k, l)$'s neighbors, pixel $I(k, l)$ will be one of pixel $I(i, j)$'s neighbors.

Cliques are associated with neighborhood structures. A set of pixel sites $c$ in $\mathcal{N}$ is a clique if all pairs of sites in $c$ are neighbors. One example is shown in Fig. 3.3. Fig. 3.3(a) shows the four-pixel neighborhood structure (NS) and Fig. 3.3(b) shows the valid associated cliques. The three-pixel clique shown in Fig. 3.3(c) is not a valid clique for the four-pixel NS in Fig. 3.3(a). Let $P_u$ represent the top pixel, $P_c$ represent the bottom left pixel and $P_r$ represent the bottom right pixel in the triangular structure. According to the four-pixel NS definition, pixels $P_u$ and $P_c$ are neighboring pixels and $P_c$ and $P_r$ are neighboring pixels. However, the pixel pair $P_u$ and $P_r$ are not declared as neighboring pixels. Thus, not all the pixel pairs in this clique are neighboring pixels and it is not a valid clique.

On the other hand, not all the valid cliques shown in Fig. 3.3(b) have to be considered in a single problem. Instead, the considered cliques are chosen according to application purposes. For example, in the Gaussian MRF model [52], only the single pixel clique in Fig. 3.3(b) is considered in many cases. Another example


Figure 3.3: The four-pixel neighborhood structure and the associated cliques.

is with the line process [53]. A line process is a member of the MRF family and it deals with discontinuity features in an image. It only considers pair-site cliques because the absolute intensity level of a single pixel could not reflect discontinuity features.

### Potential and Energy Functions

A potential function is a function $V_c$ that defines the interactions of pixel sites in clique $c$. The potential function relates to the energy function as $U(\omega) = \sum_{c \in \mathcal{C}} V_c(\omega)$. From the definition of potential and energy functions, we can see that cliques define the range of the interactions that contribute to the total energy of the whole image.

### Gibbs Distribution and the Markovian Property

The probability distribution shown in (3.3) is called the Gibbs Distribution. An interesting aspect of this probability distribution is that it maximizes the entropy (the uncertainty in the outcome) among all probability laws with given expected energy [54].

If an image is modeled as an MRF, the global probability of each configuration $\omega$ of this image is governed by the Gibbs Distribution. The Gibbs distribution describes the global probability of a configuration $\omega$ while the MRF describes the interactions of local neighboring pixels in $\omega$.

For clarity, we formally state the MRF property as follows. With the defined neighborhood structure $\mathcal{N}$, the Markovian property is described in terms of the

conditional probability distribution as

$$p(\mathbf{I}(i,j) = x|\mathbf{I}(k,l), (k,l) \neq (i,j)) = p(\mathbf{I}(i,j) = x|\mathbf{I}(k,l), (k,l) \in \mathcal{N}(i,j)), \quad (3.4)$$

where $\mathbf{I}$ represents an image and $(i,j), (k,l)$ represent pixel sites in the image. The conditional probability of a pixel given all other pixels in the image equals the conditional probability given its neighboring pixels.

The Gibbs Distribution and MRF are equivalent to each other. The formal statement is as follows.

**Statement (Hammersley - Clifford Theorem)**: A field is an MRF on $\mathcal{S}$ with respect to $\mathcal{N}$ if and only if there is a Gibbs Distribution on $\mathcal{S}$ with respect to $\mathcal{N}$. Sometimes "with respect to $\mathcal{N}$" is stated as "with respect to the nearest neighborhood $\mathcal{N}$". Here "nearest neighbor" does not mean geometrically nearest neighborhood structure, but means "nearest neighbor Gibbs potentials". Actually, both "with respect to $\mathcal{N}$" and "with respect to nearest neighbor $\mathcal{N}$" should be interpreted as follows.

**1.** Cliques are only defined on neighborhood structure $\mathcal{N}$.

**2.** Only nearest neighbor Gibbs potentials are valid. A potential $V$ is a nearest neighbor Gibbs potential if $V_c(\omega) = 0$ whenever $c$ is not in the clique set.

The proof of the statement can be found in [55–58]. The proof that a Gibbs Distribution (GD) results in an MRF is attached in Appendix B. The other direction where MRF results in a GD is also true. However, the proof for it is quite involved.

This statement has two practical values:

**1.** The joint probability $P(\omega)$ can be specified by specifying the clique potential functions $V_c(\omega)$.

**2.** $V_c(f)$ can be chosen to impose desired system behavior.

## 3.2 The Application of Markov Random Fields in Image Processings

If each pixel site in an image is assigned one value from a finite set, such an assignment is called a *configuration*. In other words, each image is treated as a 2-D random variable, where each occurrence is called a configuration.

In natural images, neighboring pixels are highly correlated with each other. This correlation can be described by power spectral densities, or correlation functions [59] or MRFs. The spectral method is a global description, which is used in Wiener filtering theory based image de-noising [60–62] and super-resolution [63,64]. The correlation function can be used both globally and locally. When used globally, it is equivalent to the spectrum methods. When used locally, it is generally used with the locally stationary Gaussian assumption [49]. In contrast, Markov methods are highly localized and can describe very complex interactions between neighboring pixels, such as smoothness and abrupt discontinuity.

The "traditional" applications of MRF models in image processings focus on wavelet domain de-noising, texture modeling and texture segmentation fields. The following subsections provide a brief review of each.

### 3.2.1 Wavelet-Domain De-noising

Based on the observation that wavelet coefficients of noise have a much weaker correlation between scales than coefficients of a clean image [65], Donoho and Johnstone proposed the wavelet shrinkage technique [66] for de-noising. In this technique, the wavelet coefficients that have an absolute value smaller than a threshold are set to zero while the magnitudes of other coefficients are reduced proportionally or non-linearly. The procedure of the shrinkage technique can be summarized as follows.

(I) Perform wavelet transformation of the noisy image.

(II) Modify the coefficients of each subband output according to:

- (1) Compute the local Hölder field of each subband. The local Hölder is a local regularity measure of the inter-scale coefficients [67].
- (2) Generate a binary mask by applying a threshold $T$ to the local Hölder field.
- (3) According to the mask result, either set the coefficients to zero or reduce their magnitude according to an estimated shrinking function.

(III) Reconstruct the clean image through inverse wavelet transform.

In the procedures above, the MRF helps with binary mask refinement [68] and producing spatially-adaptive shrinkage functions [69–71]. For binary mask refinement, an eight-pixel neighborhood MRF behaves as a regularizer on the binary mask. The applied potential function is defined as $V_{s,t} = \gamma$ with $l_s = l_t$, and $V_{s,t} = -\gamma$ with $l_s \neq l_t$, where $l_s$ and $l_t$ represent labels and $\gamma$ is a constant. A penalty is assigned if neighboring labels in the binary mask are different. Thus, through energy minimization, the mask field is smoothed and regularized.

MRF also helps in producing spatial-adaptive shrinkage functions. Traditionally, a linear shrinking function is used. For example, to restore the clean image $\mathbf{x}$ from its observation $\mathbf{y}$, the linear shrinking function is

$$x_{i,j} = \frac{\sigma_x^2}{\sigma_x^2 + \sigma_n^2} y_{i,j}, \tag{3.5}$$

where $\sigma_x^2$ represents the variance of $\mathbf{x}$ and $\sigma_n^2$ represents the variance of noise. This shrinking function assumes independent and identical Gaussian distributions of $\mathbf{x}$. This assumption disagrees with the strong non-Gaussian property that the wavelet coefficients of natural images exhibit and thus is inefficient.

By modeling the local Hölder field as a line process [53], the MRF model can estimate the conditional probability, $p(l_s|\mathbf{M})$ of each label $l_s$, where $\mathbf{M}$ represents the local Hölder field. This conditional probability is an indicator of the reliability level of each noisy pixel and can either be used directly as the shrinkage function [69] or as a factor to modify the linear shrinkage function [68].

### 3.2.2 Texture Modeling

In addition to wavelet-domain de-noising, MRF finds applications to texture modeling [72–74]. In [72], the motion texture is modeled as a parametric mixed-state auto-model. In this model, only cliques that contain no more than two pixels are considered. The major advantage of the auto-model is that its parameters can be relatively easily decided. However, two-pixel cliques are too small to capture features of natural texture, which usually have a scale larger than three or four pixels.

As a compensation, Zhu, *et al.* in [73] proposed the FRAME (Filters, Random Fields and Maximum Entropy) model for natural textures. In this model, each texture is specified by a series of filters. The modeling process is to determine the filters from a predefined filter bank and a series of linear combination factors. In this context, the texture is modeled globally as a Gibbs distribution field, where the energy function is a weighted summation of the histogram of each filtered result. This model is able to model interactions at large scales. However, this model is developed based on the assumption that any homogeneous texture can be synthesized by feeding the white Gaussian field to a series of filters. Consequently, its application is limited to homogeneous textures. Besides, the computational complexity is extremely high and there are difficulties in filter bank designs because the synthesis results are very sensitive to the filters that are available to choose from [74].

### 3.2.3 Texture Segmentation

Segmenting an image is defined as the assignment of a label to each location of an image grid [75]. Under the MAP (Maximum A Posteriori) criterion in the Bayesian estimation framework, the optimal result corresponds to a label realization $\mathbf{l} = \{l_i\}$, where $l_i \in \{1, 2, \cdots, L\}$ is the label value that maximizes the probability

$$p(\mathbf{l}|\mathbf{x}) \propto p(\mathbf{x}|\mathbf{l})p(\mathbf{l}), \tag{3.6}$$

where $\mathbf{x}$ denotes the texture image to be segmented. For MRF-based texture

segmentation, the segmented image or the labeled realization is governed by the Gibbs' Distribution. Consequently, by iteratively minimizing the energy of the segmented image or the labeled realization, the most likely segmented result is obtained. In the adaptive segmentation algorithm in [75, 76], the labeling realization $\mathbf{l}$ in (3.6) is modeled as an MRF field where only pair-site cliques $\{s, t\}$ can possibly have non-zero potentials. If the labeling at the two sites are different, $l_s \neq l_t$, the potential $V_{s,t} = 1$. Otherwise, $V_{s,t} = -1$. The high potential is a penalty of the differences of labeling between immediate neighbors. Consequently, it functions as a regularizer and smoothes the segmented result.

The probability term $p(\mathbf{x}|\mathbf{l})$ in (3.6) is treated as a multivariate Gaussian distribution, where each variable represents a class of feature and the correlations between features are specified by the covariance matrix in the Gaussian distribution. The multivariate Gaussian distribution belongs to the MRF family [52], where the interactions between pixels are specified by the covariance matrix.

Another MRF-based texture segmentation example is proposed in [77], and the purpose is to segment dynamic textures such as moving grass, foliage, sea-waves, trees and rivers, from a video sequence. In this algorithm, the segmentation is not performed on natural textures directly. Instead, it is performed on *motion textures* [72], which is built to reflect both the spatial characteristics and temporal motion of the dynamic textures in the video sequence. In this segmentation, the segmented label field, $p(\mathbf{l})$, is modeled as an eight-pixel neighborhood MRF, which behaves as a regularizer on the labeled field. The conditional probability is written as

$$p(\mathbf{x}|\mathbf{l}) = \Pi_{k=1}^{L} p(x_k|\mathbf{l}), \tag{3.7}$$

where $x_k$ represents the motion texture class $k$, and all the $L$ motion texture classes are assumed to be independent of one another. Motion texture is modeled as a parametric mixed-state auto-model [72], which is also a member of the MRF family.

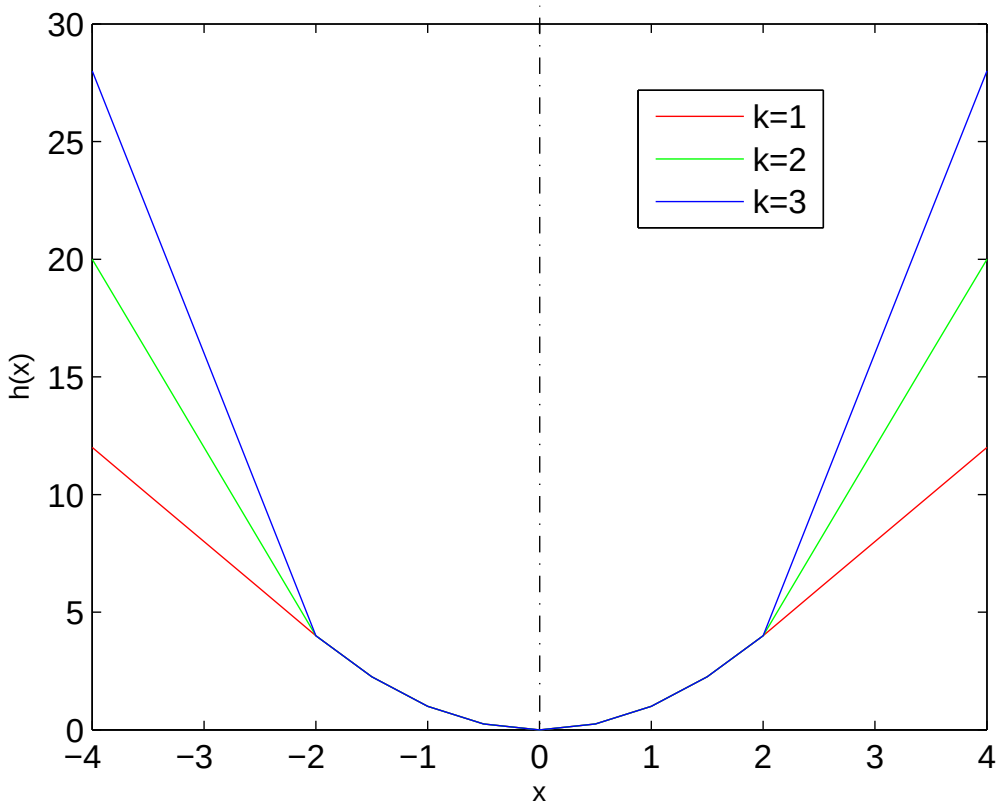



Figure 3.4: The adaptive potential function in [1].

### 3.2.4 Non-Traditional Applications

In addition to the "traditional" applications that are reviewed above, MRF is also found to be used non-traditionally in recent years, i.e., the application for blocking artifacts reduction [78], video transmission error concealment [1], and edge-preserving data detection [79]. The potential function in [1] is defined to have some adaptive properties, which uses a modified version of the traditional Huber function [80], and can be expressed as

$$h(x) = \begin{cases} x^2, & |x| < T \\ T^2 + 2kT(|x| - T), & |x| > T \end{cases} . \tag{3.8}$$

The function curve with $T = 2$ is shown in Fig. 3.4. By varying parameter $k$, the smoothing strength that is associated with magnitudes of $x$ is adjusted by

changing the slope of the matching curve. The steeper the slope is, the stronger the smoothing strength is. In the formulation in [1], the second-order pixel intensity variation is used as $x$. Thus, potentials that are associated with large second-order pixel intensity variations (PIVs) contribute more to the energy function and thus are suppressed with stronger strength. If the large PIVs represent noise, this formulation could suppress high-level noise very efficiently. However, if the large PIVs represent discontinuity features, the smoothing effect will cross edges and blur them.

## 3.3 Constraints Used in Typical MRF Models

The most critical part of an MRF model is the definition of the potential function, which imposes desired properties on the processed image. This section reviews the commonly used potential functions and their applications. Based on the knowledge of these potential functions, we propose our 2-D DAS constraint.

### 3.3.1 Gaussian MRF Model

Many MRF models use energy functions that have the following form

$$U(\omega) = \sum_{i \in \mathcal{Z}} V_1(\omega_i) + \sum_{i \in \mathcal{Z}} \sum_{i' \in \mathcal{N}_i} V_2(\omega_i, \omega_{i'}), \tag{3.9}$$

where $\omega$ denotes a random configuration of a 1-D or 2-D lattice $\mathcal{Z}$, and $\mathcal{N}_i$ represents the neighborhood structure of site $i$. Potential function $V_1$ is a single-site clique potential while function $V_2$ is a pair-site clique potential to reflect interactions between a pair of sites. Comparing (3.2) with (3.9), we can find that the energy function in the Ising model uses exactly these types of clique potentials.

In addition to the Ising model, the auto model [58] also defines the energy function based on (3.9). With $V_1(\omega_i) = \omega_i G_i(\omega_i)$ and $V_2(\omega_i, \omega_{i'}) = \beta_{i,i'} \omega_i \omega_{i'}$, where $G_i(\cdot)$ is an arbitrary function and $\beta_{i,i'}$ are constants, the energy function is

$$U(\omega) = \sum_{i \in \mathcal{Z}} \omega_i G_i(\omega_i) + \sum_{i \in \mathcal{Z}} \sum_{i' \in \mathcal{N}_i} \beta_{i,i'} \omega_i \omega_{i'}). \tag{3.10}$$

Another model that defines the energy function based on $V_1(\cdot)$ and $V_2(\cdot)$ is the Gaussian MRF model [52]. The joint probability of a multivariate Gaussian distribution is a Gibbs distribution, defined as

$$P(\omega) = \frac{\sqrt{\det(\mathbf{B})}}{\sqrt{(2\pi\sigma^2)^m}} \exp\{-\frac{(\omega-\mu)^T\mathbf{B}(\omega-\mu)}{2\sigma^2}\}, \tag{3.11}$$

where $\omega$ is the vectorized configuration, and $\mu$ is a vector containing the conditional means, and $\mathbf{B}$ is the interaction matrix, which is related to the covariance matrix by $\mathbf{B} = \sum^{-1}$ if $\sum$ represents the covariance matrix. The energy function that corresponds to the global probability distribution (3.11) is

$$U(\omega) = \frac{(\omega-\mu)^T\mathbf{B}(\omega-\mu)}{2\sigma^2}, \tag{3.12}$$

and can be expressed in terms of potential functions as

$$U(\omega) = \sum_{\omega_i,\omega_{i'}} [V_1(\omega_i) + V_2(\omega_i, \omega_{i'})], \tag{3.13}$$

where $i$ and $i'$ represent sample indices in configuration $\omega$.

$$V_1(\omega_i) = (\omega_i - \mu_i)^2/2\sigma^2 \tag{3.14}$$

and

$$V_2(\omega_i, \omega_{i'}) = \beta_{i,i'}(\omega_i - \mu_i)(\omega_{i'} - \mu_{i'})/2\sigma^2. \tag{3.15}$$

If $\mathbf{B} = [b_{i,i'}]$, $\beta_{i,i'}$ relates with $\mathbf{B}$ as follows. If the diagonal elements of $\mathbf{B}$ are unity, the off-diagonal element at $(i, i')$ is $-\beta_{i,i'}$ [81].

The MRF property of (3.11) can be expressed in terms of the conditional p.d.f. as

$$P(\omega_i|\mathcal{N}_i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\{-\frac{1}{2\sigma^2}[\omega_i - \mu_i - \sum_{i'\in\mathcal{N}_i} \beta_{i,i'}(\omega_{i'} - \mu_{i'})]^2\}, \tag{3.16}$$

where the neighborhood structure is defined in terms of the covariance matrix.

### 3.3.2 Blob and Line Processes

Both blob and line processes are proposed MRF models for image restoration. Let $\mathbf{F}$ represent a blob process and $\mathbf{L}$ represent a line process. In the blob process,

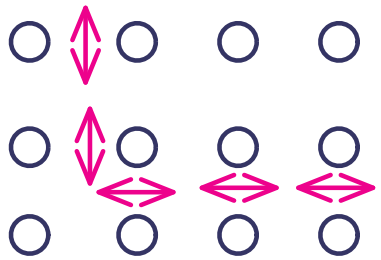


Figure 3.5: An example of a binary line process.

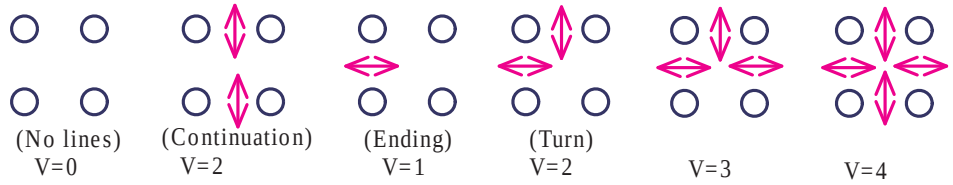



Figure 3.6: Cliques used in the line process [2].

each single pixel and its neighboring pixels are treated as blobs. The final output of each pixel leads to the minimal energy state of each blob. The blob process is observable. In contrast, the line process $\mathbf{L}$ is unobservable, which is a process that is adjoined to the blob process. Suppose the blob process that is associated with an image is on a $m \times m$ lattice. The line process will be on a "dual" $m \times m$ lattice. Each line site on the dual lattice is placed midway between each vertical or horizontal pair of pixels and represents possible edge elements. The commonly used line process is binary. That is, at each line site, there may or may not be an edge element. Fig. 3.5 shows a binary line process, where $\circ$ represents pixels and $\updownarrow$ denotes edge elements. Six line process cliques are shown in Fig. 3.6 and each clique is assigned a potential.

The major difference of these two models is the directionality. The blob process is isotropic while the line process is adaptive to local discontinuity features. Considering the widely existing discontinuity features (edges, object boundaries, etc,) in natural images, the blob process itself is not enough for image processing tasks. Instead, images are treated as the combination of blob and line processes

in practical applications, $\mathbf{I} = (\mathbf{F}, \mathbf{L})$. [2] is among the early works that use the combination of the two models to restore images from their corrupted versions. The corrupted images are modeled as

$$\mathbf{I}_{corrupt} = \Phi(H\mathbf{I}) + N \tag{3.17}$$

or

$$\mathbf{I}_{corrupt} = \Phi(H\mathbf{I}) \times N, \tag{3.18}$$

where $H$ is a lowpass filter that models the blurring process. $\Phi(\cdot)$ is absent or $\Phi(\cdot) = \sqrt{(\cdot)}$, and $N$ represents white Gaussian noise. The noise can be multiplicative or additive.

Using blob and line processes for the retrieving processing, the energy function is expressed as

$$U(\mathbf{f}, \mathbf{l}) = U(\mathbf{f}|\mathbf{l}) + U(\mathbf{l}), \tag{3.19}$$

where $U(\mathbf{f}|\mathbf{l})$ is the energy term that is associated with the blob process given the line process. If only pair-site cliques are considered in this model, the interpretation of this term is as follows. If an edge element appears between pixel sites $r$ and $s$, the connection between $r$ and $s$ is "cut-off" and the potential $V_f(f_r, f_s)$ is set to zero. Otherwise, the potential is defined as

$$V_f(\mathbf{f}) = \begin{cases} 1, & f_s = f_r \\ -1, & f_s \neq f_r \end{cases} \tag{3.20}$$

where $f_s$ and $f_r$ are the two pixels in the same pair-site clique. As for the line process energy term $U(\mathbf{l})$, only cliques that contain four line sites are nonzero. With the rotation invariance assumption, there are six distinct cliques, which are shown in Fig. 3.6. In addition, each is associated with a particular energy value, which has been designed to reflect the fact that the higher the energy, the lower the probability. In addition to the example that is shown in Fig. 3.6, there are many other combinations [2] [82].

### 3.3.3 Regularizer

To restore a vectorized image $\mathbf{f}$ from its observation signal $\mathbf{d} = \mathbf{f} + \mathbf{e}$, where $\mathbf{e}$ represents noise, the regularization formulation defines the solution $\mathbf{f}^*$ to be the

global minimization of an energy function $E(\mathbf{f})$, which is

$$E(\mathbf{f}) = U(\mathbf{d}|\mathbf{f}) + U(\mathbf{f}). \tag{3.21}$$

In (3.21), $U(\mathbf{d}|\mathbf{f})$ depicts the closeness term, which measures the energy caused by the differences between the solution $\mathbf{f}$ and observation data $\mathbf{d}$. Weighted mean square error can be used for the measurement. When $\mathbf{f}$ is continuous, $U(\mathbf{d}|\mathbf{f})$ can be defined as

$$U(\mathbf{d}|\mathbf{f}) = \sum_i (f_i - d_i)^2 \tag{3.22}$$

where $i$ represents a sample index in the solution and observation. The smoothness term $U(\mathbf{f})$, also called the *regularizer*, measures the energy caused by the irregularities of the solution $\mathbf{f}$, which in turn can be measured by the nth derivative magnitude $|\mathbf{f}^n(i)|$. It is generally defined as

$$U(\mathbf{f}) = \sum_{n=1}^{N} U_n(\mathbf{f}) = \sum_{n=1}^{N} \sum_i g(\mathbf{f}^{(n)}(i)). \tag{3.23}$$

Regularizers are different for different choices of function $g(\cdot)$. The standard quadratic regularizer has a form

$$g(\mathbf{f}^{(n)}(i)) = (\mathbf{f}^{(n)}(i))^2. \tag{3.24}$$

From (3.24), one can observe that the more irregular $\mathbf{f}^{(n-1)}(i)$ is at $i$, the larger the $|\mathbf{f}^{(n)}(i)|$. Consequently, a larger potential $g(\mathbf{f}^{(n)})$ contributes to $U_n(\mathbf{f})$. In other words, the smoothing strength in the quadratic regularizer is solely proportional to $\mathbf{f}^{(n)}(i)$. It will cause oversmoothing at discontinuities where the derivative is infinite.

Similar to the concept in blob and line processes, the smoothness constraint in the line process regularizer is removed where the magnitude of the signal derivative exceeds a certain threshold. One example can be

$$U_1(\mathbf{f}, \mathbf{l}) = \sum_{i=2}^{m} [f_i - f_{i-1}]^2 [1 - l_i] + \sum_{i=2}^{m} l_i, \tag{3.25}$$

where $l_i$ is binary, $l_i \in \{0, 1\}$, and $m$ is the total number of points on the configuration $\mathbf{f}$. Wherever a discontinuity occurs at point $i$, $l_i$=1 and the smoothness constraint is switched off.

### 3.3.4 1-D Discontinuity-Adaptive Smoothness Constraint

In the line process (LP) regularizer, the control over the smoothing strength is discrete and binary (on, off), of which the control efficiency is quite limited, whereas continuous control is more exactly adaptive to local context. Li in [3, 83] presents such a continuous adaptive regularizer, which is called the 1-D discontinuity-adaptive smoothness (DAS) constraint. The 1-D DAS constraint is adaptive to discontinuities without switching on or off. The underlying principle is to diminish the interactions between neighboring points if a discontinuity occurs.

Assuming that only the first order derivative $\eta = f^{(1)}(x)$ is involved in $g(f^{(n)}(x))$ in (3.23), Li analyzed the necessary conditions for the function $g(\eta)$ to have the continuous DA property, which can be summarized as follows.

Function $g(\eta)$ is normally chosen to be even ($g(\eta) = g(|\eta|)$), and the derivative of $g(\eta)$ can be expressed as the form

$$g'(\eta) = 2\eta h(\eta), \tag{3.26}$$

where $h(\eta)$ is called the *adaptive interaction function* (AIF) and defines the interactions between points. $g'(\eta)$ is the smoothing strength indicator. The form of (3.26) guarantees that the smoothing strength is related to the differentiation level $\eta$, and at least in a certain range, the smoothing strength can increase as $\eta$ increases.

In order to be adaptive to discontinuities, an AIF function has to satisfy

(i) $h(\eta) \in C^1$

(ii) $h(\eta) = h(-\eta)$

(iii) $h(\eta) > 0$

(iv) $h'(\eta) < 0 (\forall \eta > 0)$

(v) $\lim_{\eta \to \infty} |\eta h(\eta)| = C < \infty$

The continuity requirement in (i) guarantees the twice differentiability of the function in (3.21), which is necessary for the existence of a solution $f$ according to

Table 3.1: Four proposed adaptive interaction functions (AIFs) and the corresponding adaptive potential functions (APFs) in [3]

| Index | AIF, $h_\gamma(\eta)$ | APF, $g_\gamma(\eta)$ |
|---|---|---|
| (1) | $\exp\{-\frac{\eta^2}{\gamma}\}$ | $-\gamma \exp\{-\frac{\eta^2}{\gamma}\}$ |
| (2) | $\frac{1}{(1+\frac{\eta^2}{\gamma})^2}$ | $-\frac{\gamma}{1+\frac{\eta^2}{\gamma}}$ |
| (3) | $\frac{1}{1+\frac{\eta^2}{\gamma}}$ | $\gamma \log(1+\frac{\eta^2}{\gamma})$ |
| (4) | $\frac{1}{1+\frac{\lvert\eta\rvert}{\gamma}}$ | $\gamma\lvert\eta\rvert - \gamma^2 \ln(1+\frac{\lvert\eta\rvert}{\gamma})$ |

the Euler-Lagrange Differential Equation [84]. The evenness of (ii) is assumed for unbiased smoothing. Condition (iii) guarantees the sign of $g'(\eta)$ will not be altered by $h(\eta)$. Condition (iv) guarantees the decreasing interaction as the magnitude of the derivative increases. Condition (v) provides the DA control.

Four possible functions (parameterized by $\gamma$) are proposed based on the necessary conditions as shown in Table 3.1. The corresponding curve shapes are shown in Fig. 3.7, where one can observe that the smoothing strength is strongest when the magnitude of $\eta$ is small. As $\eta$ goes to infinity, the smoothing strength goes to zero.

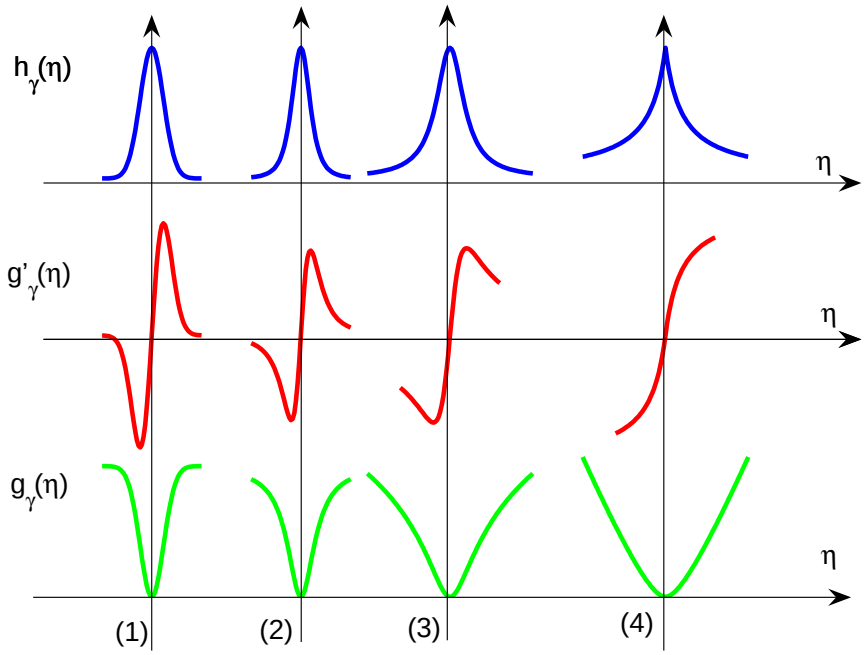



Figure 3.7: The graphic demonstration of the four DAS constraints

# 4

# Two-Dimensional Discontinuity-Adaptive Smoothness (DAS) Constraint

## 4.1 2-D Extension of the 1-D DAS Constraint

As reviewed in Section 3.3, Li in [3] studied the energy functions that could incorporate the DAS constraint in the one-dimensional case. The basic idea can be summarized as follows. Pixel intensity variations (PIVs) carry energy and a large PIV carries high energy. However, large PIVs that represent discontinuity features, e.g. PIVs across edges, should be bounded in the potential function because suppression of such energy terms would blur edges. Four functions, which could incorporate the one-dimensional DAS constraint, are proposed in [3]. One of the four functions is used as a prototype function to formulate the 2-D DAS constraint, which is

$$g(\Delta I) = -\gamma e^{-\frac{\Delta I^2}{\gamma}}, \tag{4.1}$$

where the variable $\Delta I$ represents the intensity variation and $\gamma$ is a constant parameter. The function curves for various values of $\gamma$ are shown in Fig. 4.1. It can be observed that the parameter $\gamma$ controls the shape of the curve. The DAS constraint is achieved by allowing the smoothing strength $g'(\Delta I)$ to monotonically

increase with $|\Delta I|$'s increase in a certain range. Outside that range, the measure score $g(\Delta I)$ is bounded as $\Delta I$ increases to $\infty$ and the smoothing strength decreases to zero.

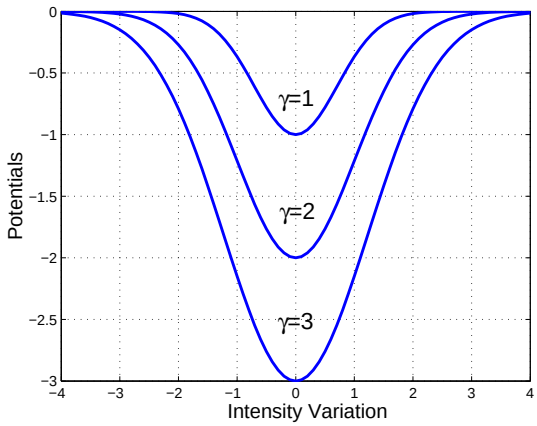



Figure 4.1: The function curves with different $\gamma$ values.

Measure scores that are calculated according to (4.1) are negative, which is not convenient for non-negative energy calculation. Based on the guideline that absolute smoothness carries no energy, $g(0) = 0$, the DAS potential function is modified as

$$V_c(\Delta I) = -\gamma e^{-\frac{\Delta I^2}{\gamma}} + \gamma, \tag{4.2}$$

which has an upper bound $\gamma$ and a lower bound 0.

In order to formulate a 2-D DAS constraint, one has to consider the edge direction information. For edges in a natural image, a discontinuity will only occur across the edge directions while strong continuity exists along the edge directions. For example, in the case of a vertical edge on a uniform background, a discontinuity occurs in the horizontal direction while strong smoothness exists in the vertical direction. High energy levels that are associated with large PIVs in the horizontal direction may represent sharpness of edges, which should not be suppressed. In contrast, high energy levels in the vertical direction represent artifacts and should be minimized. Consequently, a weighting term that indicates the directions of discontinuity features is introduced to (4.2). This bounds large PIVs independent of

direction, to formulate the 2-D DAS constraint. The potential function is expressed as

$$V_{wc}(\Delta I_k(i,j)) = w_k(-\gamma e^{-\frac{\Delta I_k^2(i,j)}{\gamma}} + \gamma), \tag{4.3}$$

and $V_{wc}$ is called the weighted potential function. $\Delta I_k(i,j)$ represents the intensity variation in the $k$th digitized direction from site $(i,j)$ and $w_k$ is the corresponding weight of this direction. The calculations of intensity variation and direction weights are presented later in this chapter.

### 4.1.1 Neighborhood Structure and Cliques

As reviewed in Subsection 3.1.2, a neighborhood structure and the associated clique structures have to be specified in order to explicitly define the potential function in (4.3).

An eight-pixel neighborhood structure is used in many algorithms, for example, traditional edge-directed interpolation [85]. However, this structure is largely chosen as a matter of convenience. It is evident that correlations between pixels extend beyond the eight-pixel neighborhood structure in a typical spatial scene. For example, only four specific digitized directions can be discriminated in an eight-pixel neighborhood structure without sub-pixel interpolation, too few to accurately portray the arbitrary edge directions of real images. A larger neighborhood can be used to define a larger number of edge directions, as illustrated in Fig. 4.2. In order to reconstruct edges with high geometric regularity, this 48-pixel neighborhood structure is used in the proposed method, which enables the sixteen digitized edge directions labeled in Fig. 4.2.

The three-pixel cliques shown in Fig. 4.3 are used as support regions of potential functions. These are more complicated than the generally used single-site and pair-site cliques. In the reviewed MRF applications, formulations rarely involve more than pairwise interactions so that the joint and conditional probabilities are relatively simple. However, such models can not reproduce large-scale spatial characteristics. For the application of enhancing edges' geometric regularity, we consider three-pixel cliques.

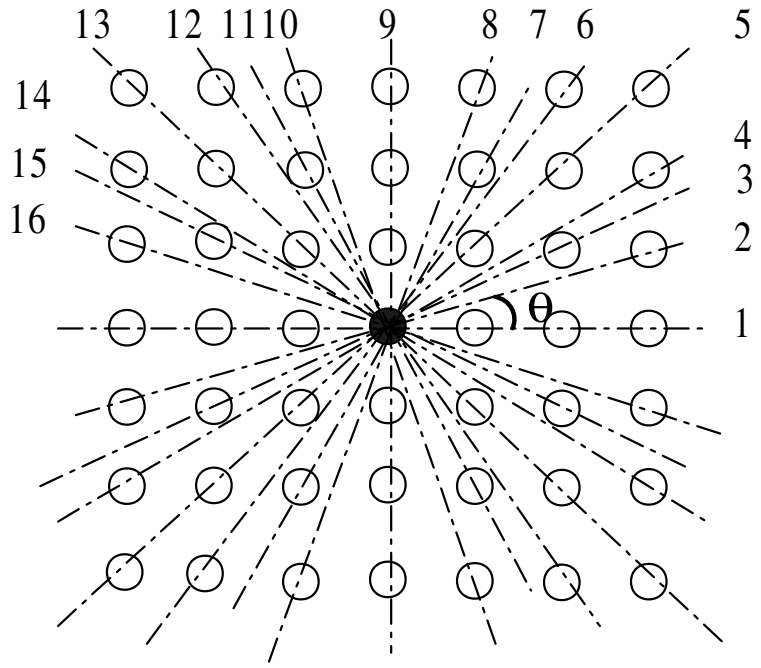



Figure 4.2: The proposed 48-pixel neighborhood structures. Circles denote neighboring pixels. Edge direction resolution $\theta$ is 18.4 degrees.

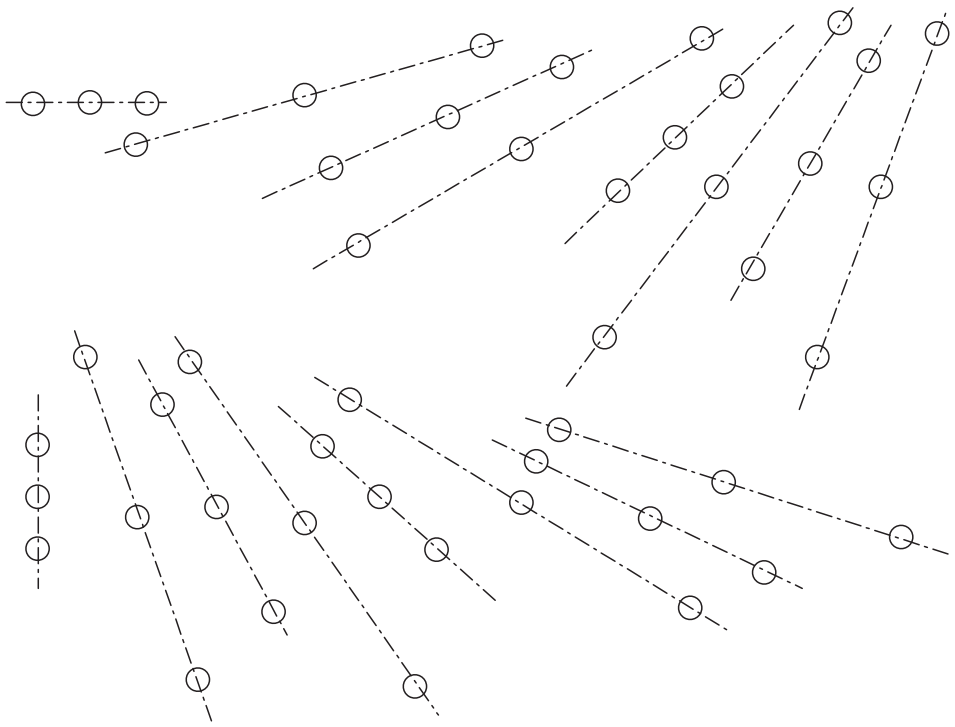


Figure 4.3: The proposed sixteen cliques.

As an example, $\Delta I_1(i,j)$ is obtained according to

$$\Delta I_1(i,j) = (|I(i-1,j) - I(i,j)| + |I(i,j) - I(i+1,j)|)/2. \tag{4.4}$$

Other $\Delta I_k(i,j)$s are obtained in a similar way, corresponding to the indices of digitized directions shown in Fig. 4.3. The $w_k$ weight term in (4.3) is found using a method presented in Section 4.2. Parameter $\gamma$ is adaptive to field data and its calculation is discussed in Section 5.3.1.

For easy reference, a *single-pixel* related energy function, $U_s$, corresponding to pixel $I$ at site $(i,j)$, is defined as

$$U_s((i,j),I) = \sum_{k=1}^{16} V_{wc}(\Delta I_k(i,j)). \tag{4.5}$$

It can be observed that $U_s((i,j),I)$ only includes the potential terms with support cliques containing site $(i,j)$. Eq. (4.5) is defined to calculate the energy difference between two configurations. These two configurations are the same except for a

single pixel site. This energy difference is defined as

$$\Delta U(I1; I2) = U_s((i, j), I1) - U_s((i, j), I2), \tag{4.6}$$

and is required for the iterative optimization procedures, which will be described in Section 5.3.2.

## 4.2 Calculation of Direction Weights for Motion Compensated De-Interlacing

The weights for the sixteen digitized directions in Fig. 4.2 are calculated in this section. The calculation process can be summarized as follows.

1) An $M \times N$ data window $\mathcal{W}$ centered at the pixel to be interpolated is chosen.

2) PIVs in various directions within the data window $\mathcal{W}$ are calculated.

3) Weights are derived based on the statistical properties of the learned intensity variations.

Details for each of these steps are presented in the following sub-subsections.

### 4.2.1 Data Window Sizes

The field to be interpolated is first temporarily de-interlaced using the "Bob" method (Linear interpolation in the vertical direction). This temporary interpolation is necessary because the pixels to be interpolated are also required for intensity variation calculation. Without interpolation, some directions will have too few intensity variation samples for weight calculation.

In implementation, we use data windows of two different sizes, $9 \times 9$ and $5 \times 5$. For windows where only one dominant edge direction is included, direction weights estimated from a larger sample set are more reliable. For textured areas, a large window will contain too many tiny edges with various directions. The direction

Table 4.1: Correspondences of single index and double indices for edge directions

| | | | |
|---|---|---|---|
| $1 \leftrightarrow (0,1)$ | $5 \leftrightarrow (-1,1)$ | $9 \leftrightarrow (-1,0)$ | $13 \leftrightarrow (-1,-1)$ |
| $2 \leftrightarrow (-1,3)$ | $6 \leftrightarrow (-3,2)$ | $10 \leftrightarrow (-3,-1)$ | $14 \leftrightarrow (-2,-3)$ |
| $3 \leftrightarrow (-1,2)$ | $7 \leftrightarrow (-2,1)$ | $11 \leftrightarrow (-2,-1)$ | $15 \leftrightarrow (-1,-2)$ |
| $4 \leftrightarrow (-2,3)$ | $8 \leftrightarrow (-3,1)$ | $12 \leftrightarrow (-3,-2)$ | $16 \leftrightarrow (-1,-3)$ |

weights that are calculated based on such a window can not reflect these very local edge directions. Consequently, the 5×5 window is used in texture areas.

In implementation, the calculation for all the pixels starts from a $9 \times 9$ data window. If the resulting direction weights imply that the most likely edge directions are not close to one another, the weight calculation will switch to the $5 \times 5$ window. With the $5 \times 5$ data window, only weights for eight directions are calculated, corresponding to directions 1, 3, 5, 7, 9, 11, 13 and 15 in Fig. 4.2.

### 4.2.2 Calculation of Intensity Variations

In this and the next subsections, discrete directions in Fig. 4.2 are also referred to with double indices, $(k, q)$, which represent the distance between the closest neighboring pixel in the direction in the top right quadrant and the central pixel (0,0). To clarify, their correspondences are listed in Table 4.2.2. Let random variable $\mathbf{\Delta I}_{(k,q)}$ represent intensity variations in digitized direction $(k, q)$. The samples of the random variable $\mathbf{\Delta I}_{(k,q)}$ are

$$\mathbf{\Delta I}_{(k,q)}(i,j) = I(i,j) - I(i+ak, j+aq), (i,j) \in \mathcal{W} \cup F_i, \max(|ak|, |aq|) = 1, \quad (4.7)$$

where $F_i$ represents the current available field, and $a$ is a scaling factor, used to satisfy that the distance between pixels $I(i+ak, j+aq)$ and $I(i,j)$ is no further than $\sqrt{2}$. For some directions, pixel $I(i+ak, j+aq)$ may have to be interpolated. This constraint guarantees different directions have the same number of samples. Otherwise, some directions would have fewer samples for the calculation of their direction weights, resulting in a biased statistical estimation.

Let the central pixel of window $\mathcal{W}$ have coordinates $(0,0)$. The range of $i, j$ is $-4 \leq i, j \leq 4$ for the 9×9 window, and $-2 \leq i, j \leq 2$ for the $5 \times 5$ window.

For direction 7, an example PIV sample can be $I(-3,1) - I(-2,\frac{1}{2})$, in which case $I(-2,\frac{1}{2})$ is bilinearly interpolated from its neighboring pixels.

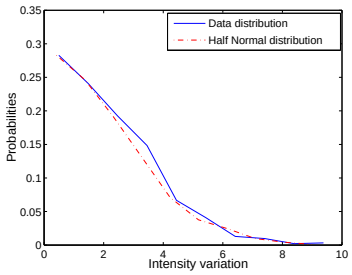
### 4.2.3 Pixel Intensity Variation Model and Direction Weights Derivation

Intuitively, PIVs in all directions of smooth regions and along edge directions of non-smooth regions are noise-like and can be modeled by zero mean Gaussian distribution. Thus, the absolute intensity variations follow the *Half Normal Distribution* model. Useful properties of the Half Normal Distribution are presented in the Appendix section 9.1. From experimental results shown in Fig. 4.4 and Fig. 4.5, we observe that the half normal distribution is an accurate probability distribution in modeling absolute PIVs in smooth areas and along edge directions in non-smooth regions. The distribution of PIVs in any directions in smooth areas or in along-edge direction is represented by solid lines in Fig. 4.4(b) and Fig. 4.5 (b). The ideal half-normal distributions are in dash lines. One can observe that, the practical and ideal curves are matched very well.

As for intensity variations associated with non-edge directions, all samples can be classified into two distributions. Those from the smooth area in the data window obey a zero mean half normal distribution and those across-edge variations obey the general (non-zero mean) half normal distribution. Direction weights are then derived from the across-edge variations. However, experiments show that the across-edge subset can contain too few samples, which hinders the correct estimation of the mean and variance. Consequently, we still use the half normal distribution to approximate the distribution of the absolute PIVs for across-edge directions.

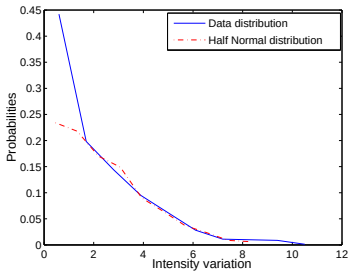
In implementation, both along- and across-edge directions are processed similarly since the zero mean half normal distribution is only a special case of the general half normal distribution. The along-edge directions will be automatically identified. The direction weight is calculated as follows.

Given $\mathbf{\Delta I}_{(k,q)} \sim N(\mu, \sigma^2)$, the new variable $\mathbf{Z}_{(k,q)} = \mu + |\mathbf{\Delta I}_{(k,q)} - \mu|$ obeys the half normal distribution. According to properties of the half normal distribution,




(a) (b)

Figure 4.4: (a) Horizontal intensity variations in the data window are calculated. (b) The data distribution is compared to the ideal half normal distribution.




(a) (b)

Figure 4.5: (a) Horizontal intensity variations in the data window are calculated. (b) The data distribution is compared to the ideal half normal distribution.

we have

$$\begin{aligned} E[\mathbf{Z}^2_{(k,q)}] &= \sigma^2 + \mu^2 + 2\lambda\sigma\mu \\ E[\mathbf{Z}_{(k,q)}] &= \lambda\sigma + \mu \end{aligned} \tag{4.8}$$

where $\lambda = \sqrt{\frac{2}{\pi}}$.

On the other hand, the absolute intensity variations $\mathbf{Z}_{(k,q)}$ can be seen as a short time realization of a wide sense stationary random process and the peak of the auto-correlation curve, $R_{(k,q)}(0)$, is adopted to specify this process. According to the definition of the correlation function and the result shown in (4.8),

$$R_{(k,q)}(0) = \mathrm{E}[\mathbf{Z}^2_{(k,q)}] = \sigma^2 + \mu^2 + 2\lambda\sigma\mu. \tag{4.9}$$

The peak values for all the sixteen directions are calculated according to (4.9), providing $\mathbf{R} = [R_1(0), R_2(0), \cdots, R_{16}(0)]$. Here, the directions are referred to using the single index notation. This vector $\mathbf{R}$ is normalized to $\hat{\mathbf{R}} = \frac{2}{\min(\mathbf{R})}\mathbf{R}$ when $\min(\mathbf{R}) \neq 0$, and $\hat{\mathbf{R}} = \mathbf{R} + 2$ when $\min(\mathbf{R}) = 0$. The direction weight vector is then computed according to

$$\mathbf{W} = \frac{1}{\log_2 \hat{\mathbf{R}}}. \tag{4.10}$$

It can be concluded that relatively large weights are assigned to along-edge directions by the inverse operation. Comparing the two matching function curves, $1/R_{k,q}(0)$ and $1/\log_2 R_{k,q}(0)$, shown in Fig. 4.6, one can observe that the logarithm operation helps to maintain weights of various directions in the same order of magnitude.

In addition, (4.9) indicates that, for a fixed $\sigma$ value, having $\mu < 0$ can lead to a larger weight than in the case of $\mu = 0$. This is not true since $\mu = 0$ is the necessary condition to achieve the largest weight. The case of $\mu = 0, \sigma = 0$ is associated with an absolute smooth area. Consequently, we substitute $|\mu|$ for $\mu$ in (4.9). From (4.9) and (4.10), one can observe that relatively large weights are assigned only when both $\sigma$ and $\mu$ are small, which correspond to along-edge directions or smooth image areas. Otherwise a small weight is assigned for cases as follows:

large magnitude of $\mu$, small $\sigma$: gradual intensity change at a constant speed

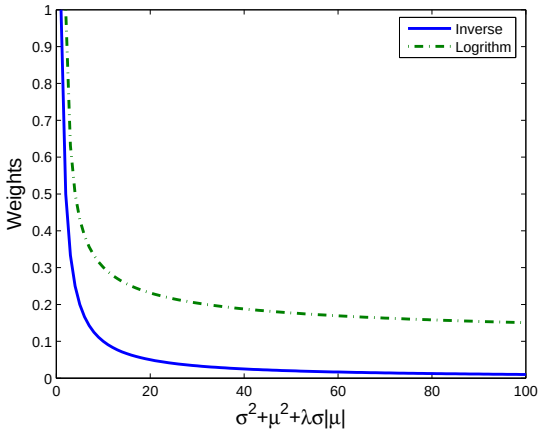



Figure 4.6: Matching function comparison.

small magnitude of $\mu$, large $\sigma$: texture areas with lots of tiny edges in all directions

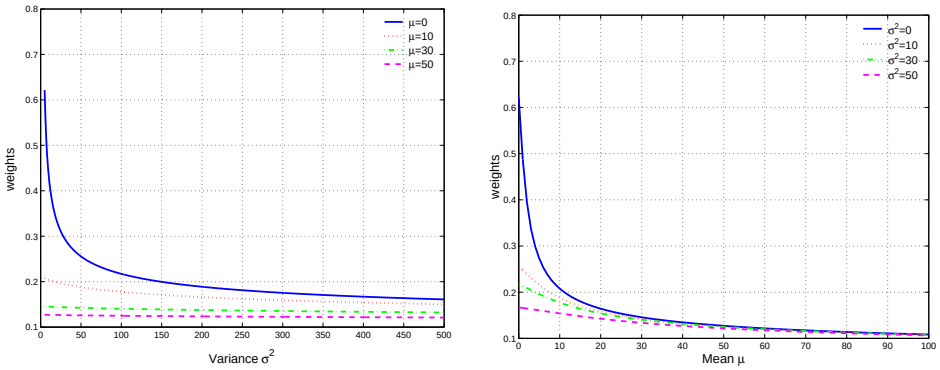
large magnitude of $\mu$, large $\sigma$: across-edge directions for an apparent edge in uniform area

The weights versus mean $\mu$ curves are shown in Fig. 4.7(a) and the weights versus variance $\sigma^2_{(k,q)}$ curves are shown in Fig. 4.7(b). It can be observed that the only case for which the weight is significant is the one for which both variance $\sigma^2$ and mean $\mu$ are small. This corresponds to cases where the local area is smooth or the direction is along-edge.

## 4.3 Edge Direction Weights Estimation for Spatial Interpolation

Edge direction weights estimation in the spatial interpolation problem follows similar procedures as in the motion compensation de-interlacing problem. However, the following modifications are made:

The direction weights are derived according to (4.12), instead of (4.9). This modification is trivial since the second order terms in (4.9) dominate the




(a) (b)

Figure 4.7: (a) The weights versus variance $\sigma^2_{(k,q)}$ curves, (b) the weights versus mean $\mu$ curves.

result. Furthermore, all the direction weights of a pixel are normalized at the final step. This modification is to simplify the calculation.

The weight calculation is performed in data windows of locally adaptable size. This modification is based on the observation that, in natural images, a pixel is correlated with its local context at different scales.

The details of the calculations are as follows. Suppose the high resolution image $H$ is related to the low resolution image $L$ as $H(2i-1, 2j-1) = L(i, j)$; We consider sixteen discrete directions for each pixel site in $H$ as indexed in Fig. 4.2. Each direction is represented by a vector $V = (v_r, v_c)$, which is the distance between the center pixel and its closest neighboring pixel in the corresponding direction. For example, direction 2 is represented by vector $(-1, 3)$. Each pixel in image $H$ can be represented by one of the four coordinates, $H(2i-1, 2j-1), H(2i-1, 2j), H(2i, 2j-1)$ and $H(2i, 2j)$. We take pixel site $H(2i-1, 2j-1)$ as an example to present the calculation of the weights. Weights for other pixel sites can be obtained in a similar manner.

First an $11 \times 11$ data window $W$, which is centered at pixel site $H(2i-1, 2j-1)$, is formed from the high resolution image $H$ and then weights are learned from PIVs in this data window. Before calculating the PIVs, we temporarily interpolate the unavailable pixels in $W$ by the bilinear method. This temporary interpolation is

necessary. Otherwise, the PIVs for some directions (e.g., direction 2, 6, 8, 10,12 and 16) are too few for the unbiased estimation of the statistical parameters. On the other hand, although bilinear interpolation blurs the original edges, main edge directions are reserved during the interpolation.

When all pixels in $W$ are available, the PIVs in all sixteen directions are calculated. Consequently, an intensity variation set for each direction is obtained as

$$\begin{aligned}\mathbf{\Delta I}(v_r, v_c) &= \{H(2(i+k)-1, 2(j+q)-1) - \\ &\quad H(2(i+k)-1+cv_r, 2(j+q)-1+cv_c)\}, \end{aligned} \tag{4.11}$$

where the range of $k, q$ is determined by satisfying the condition that $H(2(i+k)-1, 2(j+q)-1)$ is in data window $W$. Scalar $c$ is used to guarantee that $\max(|cv_r|, |cv_c|) = 2$ (equivalent to 1 in terms of the low resolution image). Eq. (4.11) shows that each pixel pair that is used for PIV calculation has at least one pixel available from the low resolution image. This guarantees the reliability of the samples used and that the same number of samples are used for mean and variance estimation for all directions.

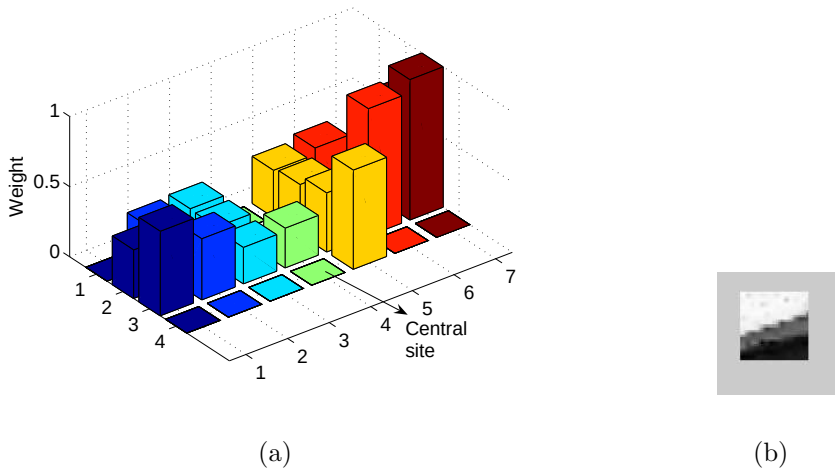
Let $\mu$ and $\sigma^2$ respectively represent the mean and variance of $\mathbf{\Delta I}(v_r, v_c)$. The weight for the discrete direction $(v_r, v_c)$ is obtained as

$$w = \frac{1}{\log_2(\sigma^2 + \mu^2)}. \tag{4.12}$$

This can be interpreted as follows. The intensity variation samples that are obtained through (4.11) can be regarded as a short time realization of a wide sense stationary random process. The auto-correlation function $R(t)$ at $t = 0$ is used to represent the process and $\sigma^2 + \mu^2$ is the estimation for $R(0)$. The inverse operation is introduced so that weights are proportional, but not inversely proportional, to continuity strengths. The logarithm operation is used to keep all the sixteen weights of a pixel site in the same order of magnitudes. Otherwise, the weights decrease too quickly with respect to $R(0)$.

In addition, to limit all the weights of a pixel site to range [0 1], the sixteen $R(0)$s of a pixel site are scaled by a factor of $\frac{2}{R_m(0)}$, where $R_m(0)$ represents the

minimal value of the sixteen $R(0)$s. Thus, the largest weight is normalized to equal one. For ideal step edges, variances and means equal to zero could be obtained along edge directions and $R_m(0)$ could be zero. In this case, all sixteen $R(0)$s would be shifted up by 2 to $R(0) + 2$. It can be concluded that the only case where the weight is significant is the one for which both variance $\sigma^2$ and mean $\mu$ are small. This corresponds to cases where the directions are along edges. Consequently, the edge direction information of a pixel is indicated by a length-$N$ vector, where $N$ is the number of considered discrete directions. Relatively large weights indicate relatively strong continuity. An example weight vector is shown in Fig. 4.8(a), where the height of a bar represents the magnitude of a weight. It can be observed that direction 2 has the largest weight. Directions 1, 3 and 4 have relatively large weights compared to across edge directions, such as 10, 11, 12, etc. This weight vector is calculated for the central pixel of a portion of the "Foreman" image as shown in Fig. 4.8(b).




(a) (b)

Figure 4.8: An example of direction weights, the direction weights shown in (a) correspond to the central pixel shown in (b).

The weight vector is a reliable indicator of local edge directions in cases where only one significant edge is in a uniform background. In contrast, for pixels in texture area, it is very likely that the data window contains multiple edges in random directions. To deal with this case, we check the top three or four most

likely directions of the sixteen directions. If they are distributed randomly rather than being close to each other, we shrink the data window and re-calculate the weights in a more localized region. In practice, three window sizes are available: 11×11, 7×7, and 3×3. The corresponding weight vectors have lengths 16, 8 and 4, respectively. This design is based on the observation that pixels in an image are correlated to their surrounding pixels at various scales.

The text of this chapter is adapted from parts of the materials that will be published as: M. Li and T. Q Nguyen, A De-Interlacing Algorithm Using Markov Random Field Model, *IEEE Transactions on Image Processing* and as: M. Li and T. Q Nguyen, Markov Random Field Model-Based Edge-Directed Image Interpolation, *IEEE Transactions on Image Processing*. The dissertation author was the primary researcher and the co-author T. Q Nguyen listed in these publications directed and supervised the research forming the basis for this chapter.

# 5

# Motion Compensated De-Interlacing

## 5.1 Introduction

Given the model-dependency and structural difficulties with field ME, an efficient protection strategy against erroneous MVs is necessary for a MC de-interlacer. However, the protection should not be over-protective, which could limit the advantages of MC.

There are deficiencies in the three existing protection strategies that are reviewed in Section 2.1.4. In median filtering protection, the median value of a pixel's MC candidate, vertically upper neighbor and lower neighbor is used as the final output. This strategy often filters out the motion compensated candidate and blurs edges. The threshold value method is not robust either since SAD or MSE levels are not reliable indicators of "true" motions. The potential problem with this method is the error propagation in the temporal dimension. Suppose an edge in the reference frame has been de-interlaced with jagged edges, this noise is very likely to be passed to the the current frame as error propagation in the temporal dimension.

It is a challenge for the three protection strategies to guarantee consistency and sharpness of edges. Moreover, the final output has to be the one that fits

its context best. By the MRF model, we impose the DAS constraint on the de-interlaced frame. The final output of each pixel site will be the one that fits its context best under the DAS constraint. The context includes neighboring pixels from both the available field and newly interpolated field. Specifically, the de-interlaced edges are expected to have strong geometric regularity (smoothness in along-edge direction and sharpness in across-edge direction [49]).

The organization of this chapter is as follows. The MAP-MRF formulation of the MC de-interlacing problem is presented in Section 5.2. Implementation details are presented in Section 5.3 and simulation results are presented and compared to other de-interlacing methods in Section 5.4.

## 5.2 The MAP-MRF Formulation of the MC De-Interlacing Problem

For ease of understanding, the following notations are used in this chapter:

$Fi_o, Fi_e$ – Represent odd and even fields respectively.

$\hat{Fr}$ – Represents the frame that is to be de-interlaced from an odd or even field.

$Fi_{refs}$ – Represents reference fields in temporal dimension.

$\mathcal{N}_{i,j}$ – Represents the defined neighborhood structure of pixel site $(i, j)$.

$\Delta I_k(i, j)$ – Represents the intensity variation in the $k$th digitized direction of pixel site $(i, j)$.

$U_s((i, j), I)$ – *Single-pixel related energy* of pixel site $(i, j)$ when the pixel value at $(i, j)$ is $I$.

$|\Delta I|$ – The absolute value (magnitude) of intensity variation $\Delta I$.

The de-interlacing process can be stated as reconstructing the missing field data given the available field data. Data or motion information from temporal reference frames or fields may be involved in the de-interlacing process. Consider the

field to be reconstructed as a multi-dimensional random variable. Its probability distribution can be expressed as

$$P(Fi_o|Fi_e, Fi_{refs}) = \frac{P(Fi_o, Fi_e|Fi_{refs})}{P(Fi_e|Fi_{refs})}. \tag{5.1}$$

Without losing generality, the available field, $Fi_e$, is assumed to be the even field and the field to be interpolated is odd. The MAP solution to (5.1) corresponds to the most likely reconstructed field

$$\begin{aligned} Fi_o^\star &= \arg\max_{Fi_o} P(Fi_o|Fi_e, Fi_{refs}) \\ &= \arg\max_{Fi_o} P(Fi_o, Fi_e|Fi_{refs}). \end{aligned} \tag{5.2}$$

Fields $Fi_o$ and $Fi_e$ comprise the reconstructed frame $\hat{Fr}$, which obeys the Gibbs Distribution. Consequently the joint posterior distribution, $P(Fi_o, Fi_e|Fi_{refs})$ can be expressed as

$$P(Fi_o, Fi_e|Fi_{refs}) = P(\hat{Fr}|Fi_{refs}) = \frac{\exp\{-U(\hat{Fr}|Fi_{refs})/T\}}{Z}. \tag{5.3}$$

From (5.3) and (5.2), it can be concluded that the MAP solution is also the one that minimizes the energy function

$$Fi_o^\star = \arg\min_{Fi_o} U(Fi_o, Fi_e|Fi_{refs}). \tag{5.4}$$

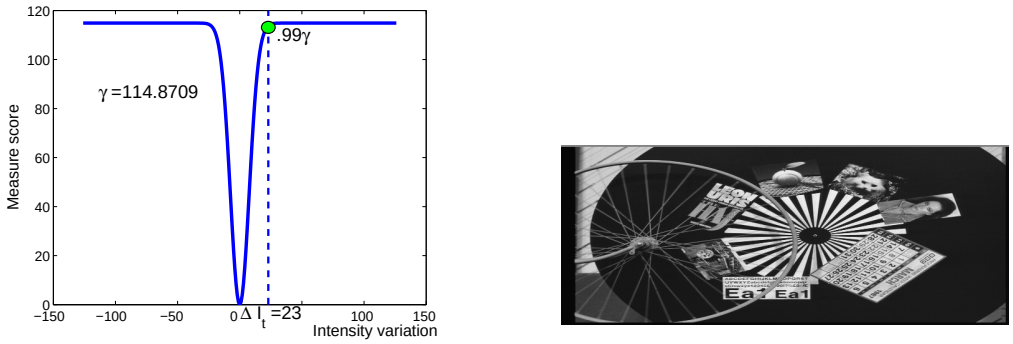
## 5.3 Implementation

### 5.3.1 Parameters Computation

The parameter $\gamma$ in (4.2) is learned from the current field data prior to the de-interlacing process. The steps are

Step 1: PIVs in the horizontal direction are calculated. Here $|\Delta I_h|$ denotes the absolute values (magnitudes) of intensity variations and $|\Delta I_p|$ denotes the top 10% samples in the data set $|\Delta I_h|$.

Step 2: Assume that the top 10% intensity variations $|\Delta I_p|$ represents significant discontinuity features in the field and let $\Delta I_t$ represent the minimal




(a) Function curve

(b) Used field

Figure 5.1: Parameter calculation example: $\gamma = 114.8709$ and $\Delta I_t = 23$.

value in $|\Delta I_p|$. Parameter $\gamma$ is determined by specifying a near-bound measure score corresponding to variation $\Delta I_t$, for example, $g(\Delta I_t) = .99\gamma$ and thus $\gamma = -\Delta I_t^2 / \ln(0.01)$.

An example calculation is shown in Fig. 5.1(a) and the field data shown in Fig. 5.1(b) is used. The same set of parameters is used for cliques of all the other fifteen directions in Fig. 4.3.

Another important parameter to be decided is the convergence speed control factor $T$. It is chosen empirically according to the energy measure metric and the desired maximal iteration number. An initial value $T_0$ is used and updated according to a decreasing rule, for example, $T_q = (\alpha)^{\lceil \frac{q}{K_s} \rceil} T_0$, where $\lceil x \rceil$ represents the minimal integer that is not smaller than $x$ and $q$ represents the $q$th iteration. $K_s$ denotes the number of iterations that run with the same $T$. $\alpha$ is a constant factor between 0.9 and 1. $\alpha$ controls the decreasing rate of $T$ and forces a faster convergence speed after every $K_s$ iterations. The specification of $T$ is the same for various video sequences.

### 5.3.2 Implementation Procedure

The proposed algorithm is implemented using the Simulated Annealing algorithm [86, 87]. The whole process searches for the MAP configuration from the state space $\mathcal{S}$, which includes all possible configurations. The state space is gen-

erally large. To decrease its size, we propose a candidate set for each pixel to be de-interlaced.

### The Candidate Set

With the 48-pixel neighborhood structure, spatial candidates are proposed as follows. First, the field is de-interlaced using the spatial linear interpolation method to make all the pixels available. Then for each of the sixteen directions, the linear average of the closest neighboring pixel pair is one member of the candidate set as shown in Fig. 5.2. Counting the corresponding MC candidate, each pixel to be interpolated would have a size-17 candidate set. If any two of the seventeen candidates are the same, only one of them is kept. Candidates in a candidate

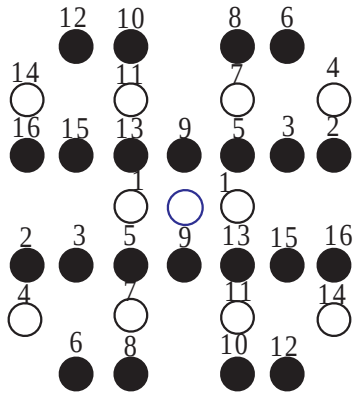



Figure 5.2: Formulation of the candidate set.

set are initialized to have equal probabilities. The equal probabilities are used to initialize the symmetric transition matrix $\mathbf{Q}$ of the Markov Chain in the Simulated Annealing algorithm. For example, $\mathbf{Q}$ is initialized as $\mathbf{Q} = [1/2, 1/2; 1/2, 1/2]$ for candidate set $\{a1, a2\}$ with $P(a1) = P(a2) = 0.5$.

### The Iterative Optimization

Step 1: Initialization of the de-interlaced frame. Theoretically, the global optimal result can be obtained independent of the starting state. However, many it-

erations are required to achieve the global optimal state. We use the de-interlaced frame with the spatial linear interpolation method as the initial state.

Step 2: At each iteration, a new candidate for each missing pixel is generated by a Gibbs sampler according to the transition probability matrix $\mathbf{Q}$, which is updated in Step 3 in each iteration. The matrix $\mathbf{Q}$ specifies the transition probabilities from each state to all other states in the candidate set.

Step 3: Suppose one candidate (new state) $s_j$ for site (i, j) is chosen from Step 2, of which the old state is $s_i$. The new state is accepted according to probability $P_\alpha = \min(1, \frac{P_{s_j}}{P_{s_i}})$, where $P_{s_j}$ and $P_{s_i}$ denote the global image's probabilities at states $s_j$ and $s_i$ respectively. From (5.3), the probability ratio relates the energy function as $\frac{P_{s_j}}{P_{s_i}} = \exp\{\frac{U_{s_i} - U_{s_j}}{T}\}$. Note that if only one pixel gets updated in a whole image, the whole image's state is completely determined by the state of this single pixel.

Eq. (4.6) is used to calculate the energy change $\Delta U = U_{s_i} - U_{s_j}$. The transition probabilities are updated according to

$$P_{s_i,s_j} = \begin{cases} P_{s_i,s_j} \min(1, \frac{P_{s_j}}{P_{s_i}}) & s_j \neq s_i \\ 1 - \sum_{k,k\neq i} P_{s_i,s_k} \min(1, \frac{P_{s_k}}{P_{s_i}}) & s_j = s_i \end{cases} \tag{5.5}$$

**Convergence Property and Complexity**

The transition matrix $\mathbf{Q}$, which is updated according to (5.5), remains irreducible symmetric during the iterative optimization process and thus the Markov Chain is irreducible and reversible [86]. Consequently, the convergence to an equilibrium probability is guaranteed.

The convergence speed and whether the iterative process converges to the global minimal energy state is associated with parameter $T$. In [2], it states that if the temperature $T(k)$ used in executing the $k$th site replacement satisfies the bound $T(k) \geq \frac{C}{\log(1+k)}$ for every $k$, where $C$ is a constant independent of $k$, the configuration generated by the iterative optimization process will be the global minimal energy state. However, the value $C$ is far too large for practical computation. The proposed de-interlacing algorithm does not guarantee global optimal solutions.

As for the complexity, we lower it by discriminating edge pixels from non-edge

pixels. Each pixel to be interpolated is labeled as either an edge pixel or a non-edge pixel. Suppose a pixel has discrete direction k (of the 16 defined discrete directions), if its two closest neighbor pixels in direction k have the same or similar directions, k or k$\pm$1, this pixel is labeled as an edge pixel. Otherwise, it is labeled as a non-edge pixel. Non-edge pixels are interpolated by the "Bob" method. Edge pixels are interpolated under the DAS constraint. The "Bob" method might not be the optimal interpolation method for non-edge pixels, but it is better for non-edge pixels to guarantee the local continuity of pixel intensities than to impose the DAS spatial constraint. Isolated intensity levels will be perceived as noise.

In addition, the complexity can be decreased greatly by implementing the Gibbs sampler with multiple processors. The Gibbs sampler in the Simulated Annealing algorithm is highly parallelizable. Currently we used a raster scan serial version of it in the iterative optimization process, which is relatively slow. The execution time can be sped up by a factor of two or three if two or three processors run simultaneously. The full parallel potential can be explored by assigning a processor to each pixel to be interpolated. A limited amount of communication among processors is required.

## 5.4 Simulation Results and Discussions

The proposed MC de-interlacing algorithm is compared to other methods in this section. In addition, the efficiency of the MRF-based protection strategy is analyzed.

### 5.4.1 De-Interlacing Performance Comparison

4-field 3DRS and GME models are used for ME. The ME block size is $4 \times 8$ in the 3DRS model and in foreground object ME in the GME model. Considering the diversity of edges in the test video sequences, we used a 48-pixel neighborhood structure in the implementation of the proposed algorithm. The size of the data windows is either $9 \times 9$ or $5 \times 5$, of which the center is the pixel to be interpolated. The convergence control factor $T$ has an original value $T_0 = 100$ and is updated

according to $T_q = (0.95)^{\lceil \frac{q}{7} \rceil} T_0$, where $q$ represents the $q$th iteration. The maximum iteration number is 100. The methods used for comparisons are abbreviated as follows.

MRF: the proposed MC de-interlacing method.

MA: Motion adaptive de-interlacing method using 4-field 3DRS ME.

Direct MC: Direct motion compensated de-interlacing method. The de-interlaced frame reflects the reliability of the MVF.

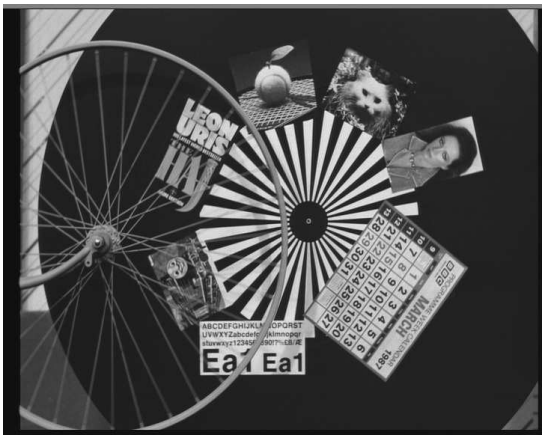
EDI: An edge-directed interpolation method suggested in [31] as a block-based directional edge interpolation method. In this method, a neighborhood structure of size 7×9 is used. Compared to other EDI methods, this method enables higher edge direction resolution.

AR: Motion compensated de-interlacing with the adaptive recursive method as a protection strategy [34].

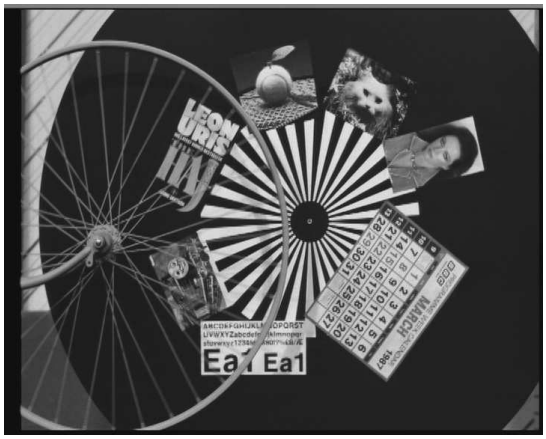
The de-interlaced video sequences are available at `http://videoprocessing.ucsd.edu/~minli/TranIPdemo.htm` in avi format.

**Example 1**: The "Bicycle" sequence is used in this example. The major motion in this sequence is rotation. Two wheels with many spikes rotate at different speeds counter-clockwise and overlap partially. The translational motion model, 4-field 3DRS, is used for ME. This is a difficult sequence for 3DRS. The comparison of the de-interlaced third frame is shown in Fig. 5.3 and the zoom in comparison is shown in Fig. 5.4. Compared to other MC de-interlacing method and the EDI method, the proposed method produces the best de-interlaced frame with sharp smooth edges.

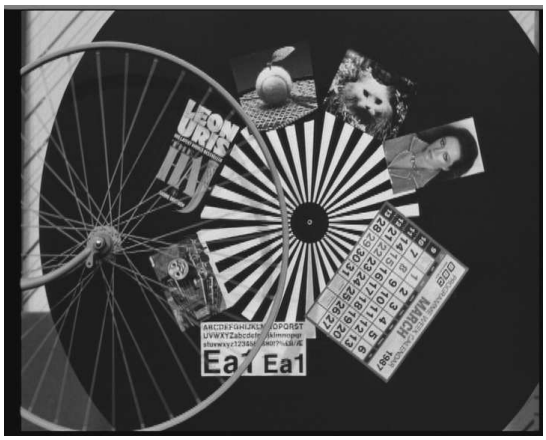
The near-horizontal edges have not been de-interlaced as well as edges in other directions. This is because these directions are beyond the edge direction resolution of the 7×7 neighborhood structure. A larger neighborhood structure will enable higher resolution of edge directions and can be expected to perform better with these edges.



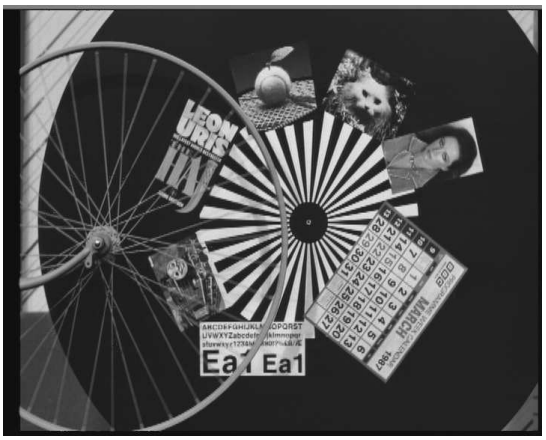
(a) Original frame



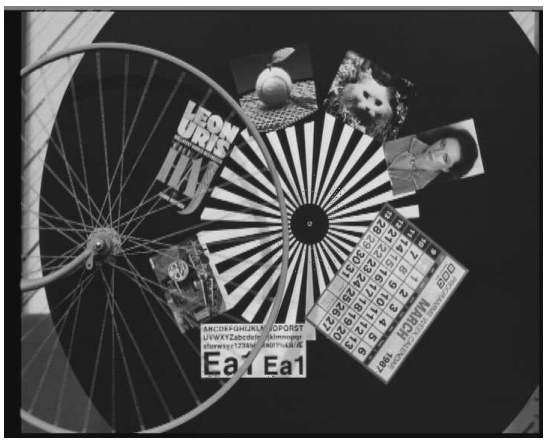
(b) MRF method



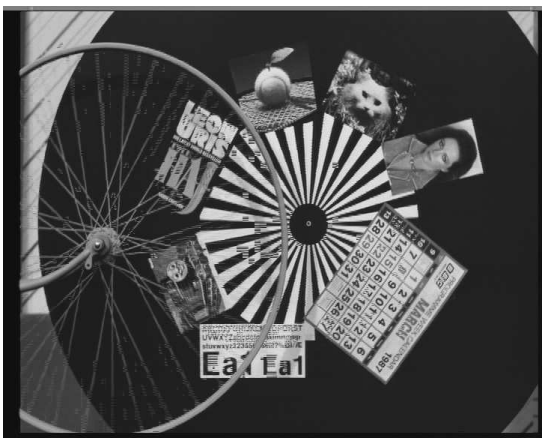
(c) AR method



(d) MA method

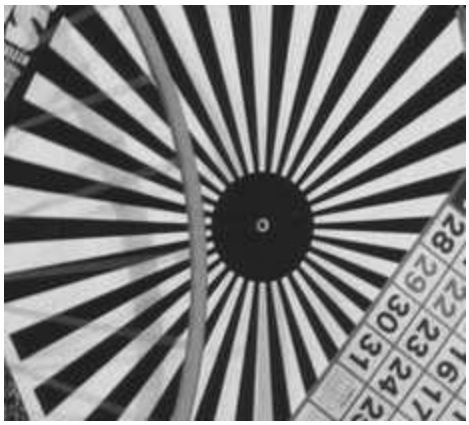


(e) EDI method

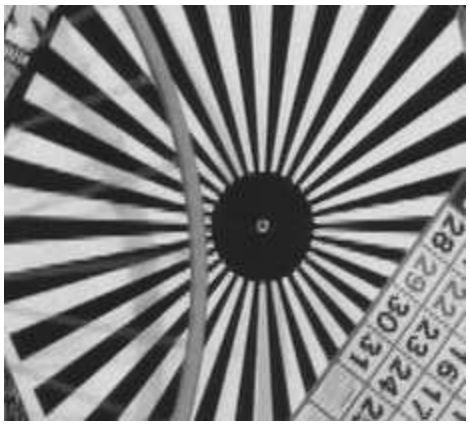


(f) Direct MC method

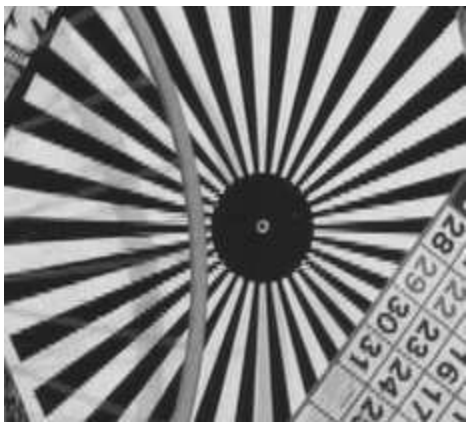
Figure 5.3: De-interlaced results of the 3rd field in the "bicycle" sequence.



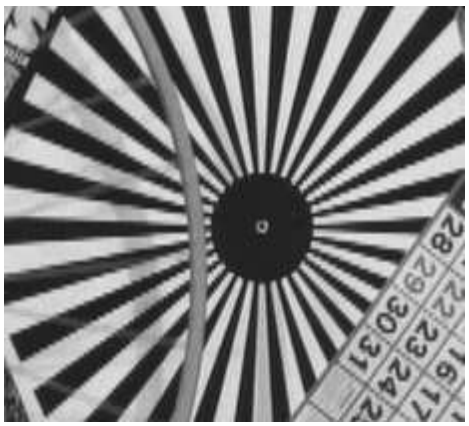
(a) Original frame



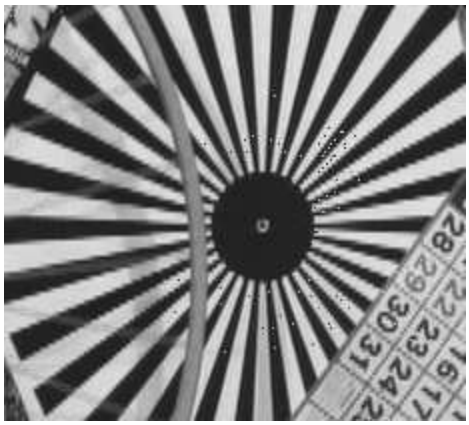
(b) MRF method



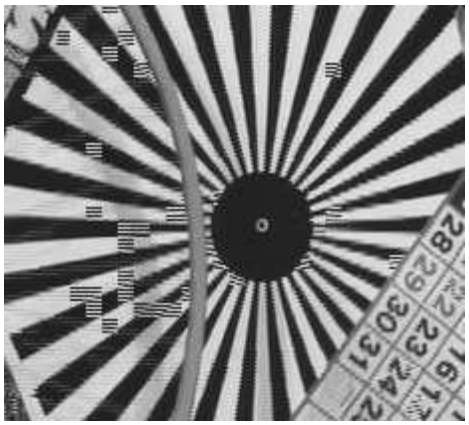
(c) AR method



(d) MA method



(e) EDI method



(f) Direct MC method

Figure 5.4: Zoom in comparison.

**Example 2**: The "Football" sequence is used in this example. The same parity GME model [30] is used to estimate the motion. The major motion in this sequence consists of two parts. One part is the global background motion caused by camera panning and zooming, the other part has several athletes as foreground objects, which are small and have different motions from the background. This model fits well with the motion in this sequence. The background motion is described by the affine matrix that is produced from phase correlation GME. As outliers, the motions of the foreground objects are estimated by bi-directional local block-matching search.

The de-interlaced results are shown in Fig. 5.5 and the zoom-in comparisons are shown in Fig. 5.6. Significant improvements with major edges can be observed.

**Example 3**: The sequence is "Foreman" and 4-field 3DRS ME is used for this sequence. From the direct MC result shown in Fig. 5.7(f), one can observe that this ME model fails in the head area, which has rotating motion. This is because this model assumes motion among the four fields is unidirectional and at constant speed. Obviously, the head turning motion does not satisfy this assumption. The sixth de-interlaced frame is shown in Fig. 5.7 and the zoom in comparisons are shown in Fig. 5.8. In comparison, the result with the proposed method is neat and edges have strong geometric regularity.

**Example 4**: This example is about the "Flag" sequence and the 4-field 3DRS model is used for ME. For this sequence, the progressive format is not available. The motion of the flag in this sequence is rather irregular. Fig. 5.9(f) illustrates the areas where the ME actually fails. The performances of various methods are presented in Fig. 5.9 and the zoom in comparisons are shown in Fig. 5.10. Subjectively, the proposed method produced much smoother edges.

### 5.4.2 Motion Compensation Maps

The maps shown in Fig. 5.11 are generated as follows. For the de-interlaced frames shown in Example 1 to 4, we label the final output of each pixel site in the MRF method. If the final output is the MC candidate, we label this site on the map as 1. Otherwise, this site is labeled as 0. Thus, the bright pixels in the maps

(a) Original frame

(b) MRF method

(c) AR method

(d) MA method

(e) EDI method

(f) Direct MC method

Figure 5.5: De-interlaced results of the 11th field in the "football" sequence.

(a) Original frame

(b) MRF method

(c) AR method

(d) MA method

(e) EDI method

(f) Direct MC method

Figure 5.6: Zoom in comparison.

(a) Original frame

(b) MRF method

(c) AR method

(d) MA method

(e) EDI method

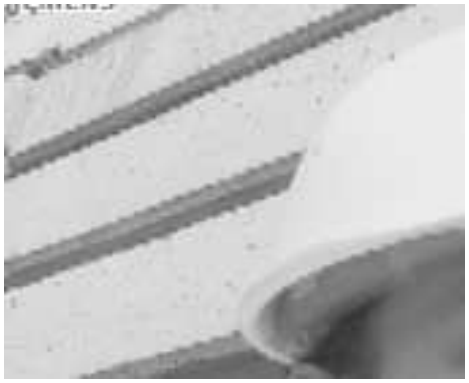
(f) Direct MC method

Figure 5.7: De-interlaced results of the 6th field in the "Foreman" sequence.

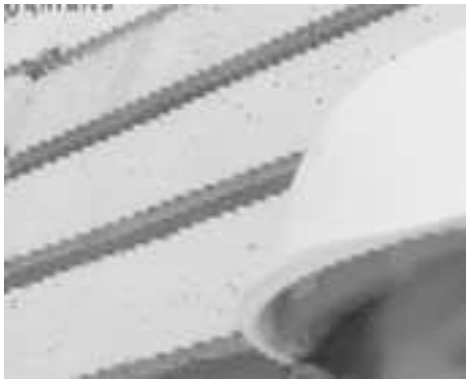
(a) Original frame

(b) MRF method

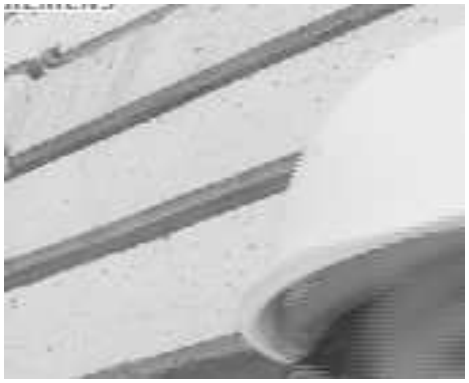
(c) AR method



(d) MA method

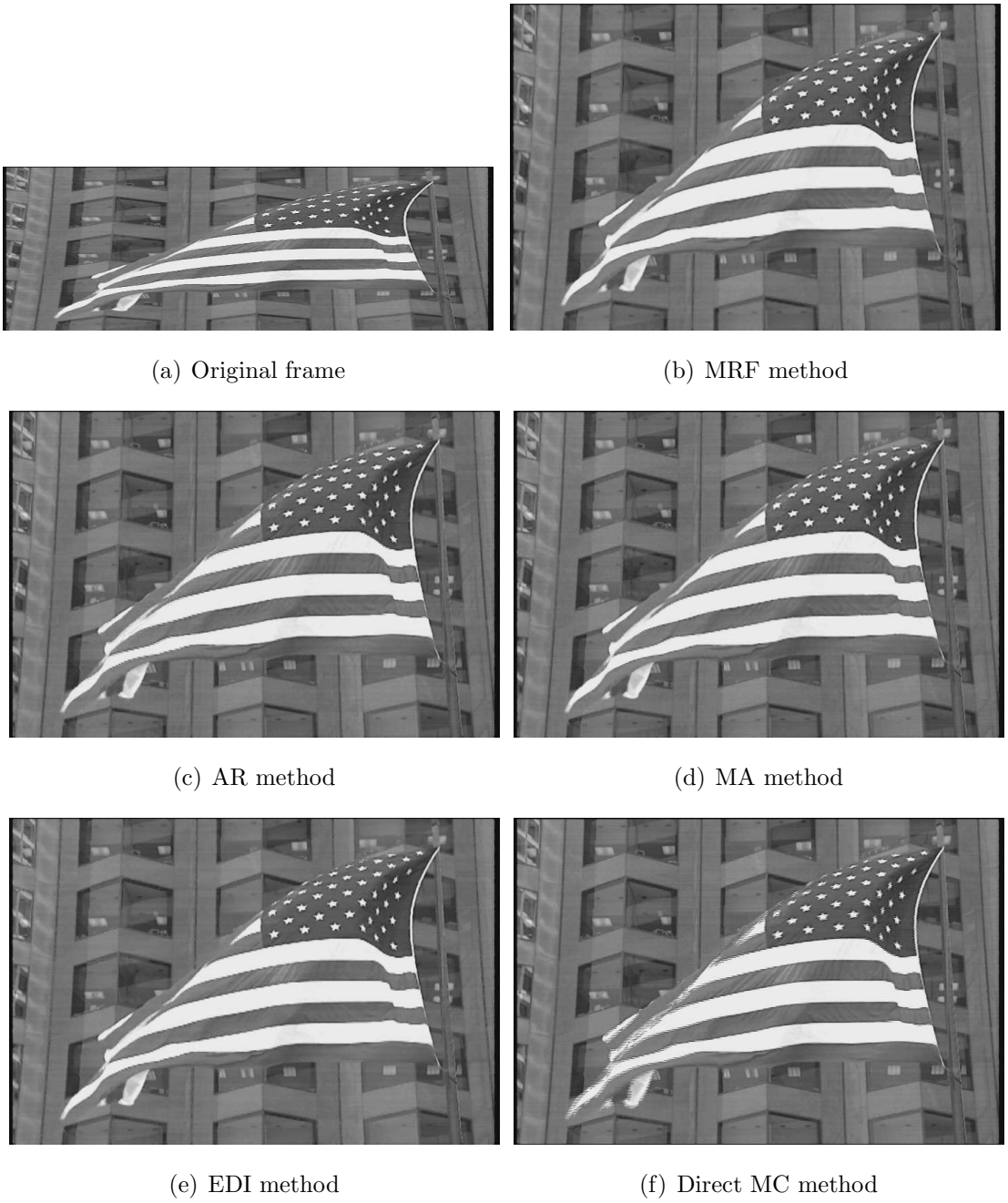


(e) EDI method



(f) Direct MC method

Figure 5.8: Zoom in comparison



(a) Original frame

(b) MRF method

(c) AR method

(d) MA method

(e) EDI method

(f) Direct MC method

Figure 5.9: De-interlaced results of the 9th field in the "Flag" sequence.

(a) Original frame

(b) MRF method

(c) AR method

(d) MA method

(e) EDI method

(f) Direct MC method

Figure 5.10: Zoom in comparison.

indicate MC candidates.

From the maps, we can observe that, in accordance to the distribution of reliable MVs, which can be observed from the direction MC results, fewer MC candidates are adopted in areas where the MVs are incorrect, such as the "Foreman"'s face area and the "Flag"'s left bottom corner. The maps show the efficiency of the MRF-based protection strategy. This strategy can be used to protect other MC de-interlaced algorithms, as a substitution for median filtering or threshold value method.

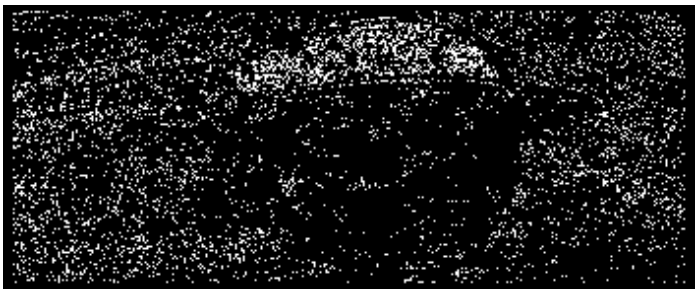
### 5.4.3 Scene Change Performance

We design an artificial sequence to test the effectiveness of the MRF-based protection strategy. The four fields from the "Foreman" sequence that are shown in Fig. 5.12 are used for 4-field 3DRS ME, the de-interlaced results of the third field are shown in Fig. 5.13. From the direct MC de-interlaced result shown in Fig. 5.13(f), we know that the estimated MVs are not reliable. In extreme scene change cases, the MRF-based protection strategy performs robustly.

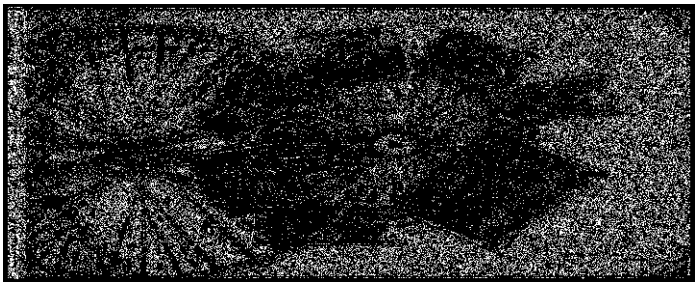
### 5.4.4 PSNR Comparison

Although Mean Square Error (MSE) is not a convincing assessment criterion of reconstructed image quality, Peak SNR (PSNR) is used as the objective measure criterion to measure the quality of the de-interlaced frames in the simulations. The average PSNR values from five different methods are plotted in Fig. 5.14. In terms of PSNR comparison, the proposed method outperforms all the other methods for the "Foreman" sequence. For "Bicycle" sequence, the PSNR level is slightly lower than the MA method. MA method converts to the "Bob" method with unreliable motion. The reconstructed edges in the MRF-based method have much higher geometric regularity. As for the "Football" sequence, the proposed method does not yield the highest PSNR level. In this sequence, a significant part is homogeneous textures, leaves and grass. In these areas, the detected local edge direction information has not been very helpful for de-interlacing. The proposed

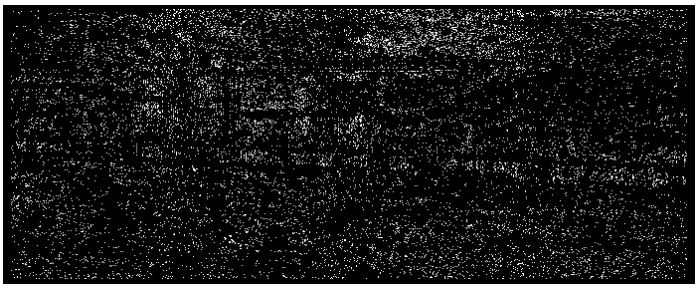
(a) Flag



(b) Foreman

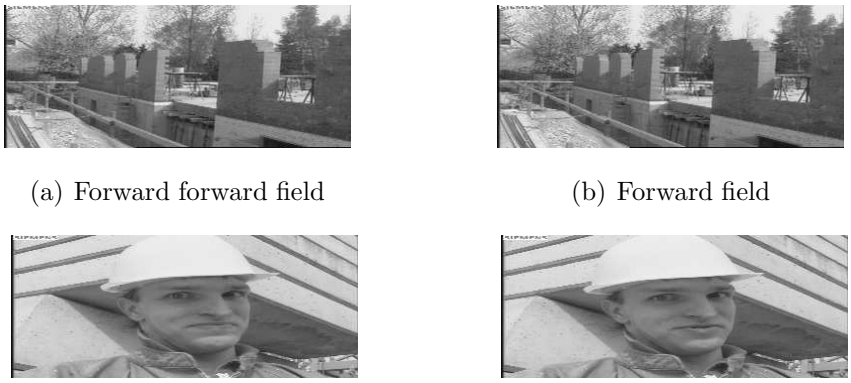


(c) Bicycle



(d) Football

Figure 5.11: Motion compensation maps.



(a) Forward forward field (b) Forward field

(c) Current field (d) Backward field

Figure 5.12: 4 fields for 3DRS ME.

method produces much sharper and more consistent edges.

## 5.5 Conclusions

In this chapter we propose a novel motion compensated de-interlacing algorithm. In the proposed algorithm, the de-interlacing process is formulated as an MAP-MRF problem. The highest probability solution is the configuration that is in a minimal energy state. By defining the energy function in the Gibbs-MRF model carefully, a minimal energy state is related with images that have highly geometric regular edges. A novel weighting scheme is developed to indicate the directions of local discontinuity features (edges) during the formulation of the DAS constraint. In comparison with traditional edge direction detection algorithms, the proposed one is robust and can decide edge directions at high resolution. Compared to other motion compensated de-interlacing algorithms, the protection strategy against erroneous MVs in the proposed algorithm is local statistical-based and should be more robust. The proposed algorithm is implemented using the Simulated Annealing algorithm. Simulation results are presented to verify the efficiency of the proposed algorithm.

The text of this chapter is adapted from material that has been accepted for publication as: M. Li and T. Q Nguyen, *A De-Interlacing Algorithm Using Markov Random Field Model*, *IEEE Transaction on Image Processing* and also in part ap-

(a) Original frame

(b) MRF method

(c) AR method

(d) MA method

(e) EDI method

(f) Direct MC method

Figure 5.13: Performance with scene change.

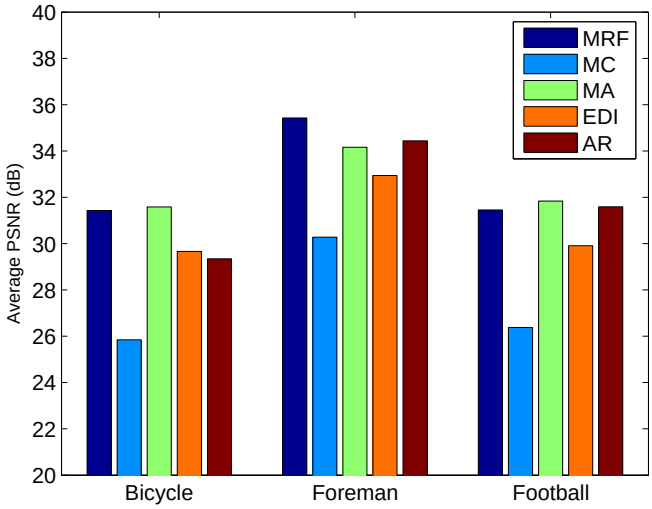



Figure 5.14: Average PSNR comparison of the de-interlaced sequences.

pears as: M. Li and T. Q Nguyen, *Discontinuity-adaptive de-interlacing scheme using Markov Random Field model*, *IEEE International Conference on Image Processing* 2006. The dissertation author was the primary researcher of this publication, and the co-author listed directed and supervised the research which forms the basis for this chapter.

# 6

# Spatial Interpolation

## 6.1 Introduction

As reviewed in Section 2.2, the NEDI method shows improvements of interpolated edges in comparison to the conventional bilinear, bicubic and spline interpolation methods. However, as a result of the fact that the interpolation of a pixel has been limited within its four nearest neighbors along the diagonal directions, the sharpness of the interpolated edges is not comparable to the original edges. Another limitation of this algorithm is that it has difficulties in dealing with texture areas. Spurious minor edges are observed in the interpolated texture areas, which make the interpolated images look unnatural. As a result of losing fidelity of the original image, the interpolated image in the NEDI method will have a lower PSNR level compared to the conventional methods if texture is contained in the image. In this chapter, we propose the MRF-EDI method to improve edges' geometric regularity further while maintaining the fidelity of the original image.

Our proposed MRF model-based edge-directed interpolation method (MRF-EDI) is an implicit edge-directed interpolation method. In MRF-EDI, the edge directions of an edge pixel are indicated by the continuity strengths in all directions. Instead of labeling each direction as either edge or non-edge, we measure the continuity strength in each direction and represent it by a rational number between 0 and 1. Large values indicate strong continuities (along edge directions) while small values indicate weak continuities (across edge directions). These values are

derived from the statistical properties of pixel intensity variations (PIVs) in a local data window. The relative continuity strengths of all directions are used as edge direction information to formulate the geometric regularity (GR) spatial constraint, which can be summarized as smoothness along edge directions and sharpness across edge directions.

The organization of this chapter is as follows. The MAP-MRF formulation of the interpolation problem is presented in Section 6.2. Implementation details are presented in Section 6.3 and simulation results are presented and compared to other interpolation methods in Section 6.4.

## 6.2 The MAP-MRF Formulation of the Spatial Interpolation Problem

As many others do in the spatial interpolation problem, we related the interpolated high resolution image to the low resolution image as $\mathbf{h}(2i-1, 2j-1) = \mathbf{l}(i, j)$, where $\mathbf{h}$ represents the high resolution image and $\mathbf{l}$ represents the low resolution image. Let $\mathbf{h}_{-l}$ represent the pixels in $\mathbf{h}$ that are not available in $\mathbf{l}$. Then the interpolation problem is to find the optimal result $\mathbf{h}_{-l}$ given the low resolution image $\mathbf{l}$. Adopting the MAP criterion in the Bayesian estimation framework, the probability of the result $\mathbf{h}_{-l}$ can be expressed as

$$p(\mathbf{h}_{-l}|\mathbf{l}) = \frac{p(\mathbf{h}_{-l}, \mathbf{l})}{p(\mathbf{l})} = \frac{p(\mathbf{h})}{p(\mathbf{l})}. \tag{6.1}$$

Given a low resolution image $\mathbf{l}$, the global probability of $\mathbf{h}$ is governed by the Gibbs distribution, which is

$$p(\mathbf{h}) = \frac{\exp\{-U(\mathbf{h})/T\}}{Z}, \tag{6.2}$$

where $\mathbf{h}$ represents one possible interpolation result. Thus the most desired interpolation result, $h^\star$, is the one that maximizes the probability $p$ and, equivalently, minimizes the energy function $U(h)$ given the low resolution image. It is

$$h^\star = \arg\min_h U(\mathbf{h}), \tag{6.3}$$

and $U(\mathbf{h})$ is the summation of the single-pixel related energy (4.5) throughout the overall interpolation result.

## 6.3 Implementation

### 6.3.1 Parameter $\gamma$ Calculation

The parameter $\gamma$ in (4.3) is derived from the low resolution image. Its calculation can be summarized in the following two steps.

**1.** PIVs in the horizontal direction are calculated. Let $|\Delta I_h|$ denote the data set consisting of the magnitudes of all the PIVs, and $|\Delta I_p|$ denote a subset, in which the magnitudes of the PIV samples are among the top 10% of the data set $|\Delta I_h|$.

**2.** Assume that $|\Delta I_p|$ represents significant discontinuity features in the image and let $\Delta I_t$ represent the minimal magnitude in set $|\Delta I_p|$, then the parameter $\gamma$ is determined by relating a close-bound measure score to the intensity variation $\Delta I_t$, i.e.,

$$-\gamma e^{-\frac{\Delta I_t^2}{\gamma}} + \gamma = .99\gamma. \tag{6.4}$$

### 6.3.2 Schematic Description of the Implementation

The steps of the MRF-EDI method can be summarized as follows.

**1.** Weight calculation according to the procedures developed in Section 4.3.

**2.** Discrimination of edge/non-edge pixels.

**3.** Proposing candidate sets for edge pixels.

**4.** Single-pass implementation or iterative optimization (using the Gibbs sampler [86] and the simulated annealing method [2]).

Details of steps 2, 3 and single-pass implementation in step 4 are presented separately below. Concrete operations in iterative optimization in step 4 is available in [88].

### 6.3.3 Discrimination Between Edge and Non-Edge Pixels

To lower the complexity, only edge pixels are optimized iteratively while non-edge pixels are interpolated using deterministic interpolation methods, for example, bicubic. A pixel is identified as an edge pixel if only its two closest neighboring pixels in its strongest continuity direction (indicated by the largest weight) have similar strongest continuity directions. For example, if the strongest continuity of the being-checked pixel site happens on direction $d$, its two closest neighboring pixels in direction $d$ must have their strongest continuities in directions $d$ or $d\pm1$ in order for this pixel to be declared as an edge pixel. Both edge and near-edge pixels will be identified as edge pixels and are optimized iteratively. Non-edge pixels include pixels in smooth areas and pixels with very short edges. For pixels in smooth areas, the MRF-EDI method is not used since no edges are reconstructed. For pixels with very short edges, where the local intensity variation is random, it is more important to guarantee local intensity continuity than to impose the DAS constraint. This is because weights calculated for these pixels are unreliable.

### 6.3.4 Proposing Candidate Sets

A candidate set for each edge pixel to be interpolated is proposed. With the proposed candidate set, each pixel to be interpolated can only have values from its candidate set, instead of any values between 0 and 255. Thus the size of the state space that the minimal energy state is searched from reduces significantly. The candidate set is proposed based on the low resolution image and, basically, one candidate for each discrete direction is proposed. Thus, including one bicubic interpolation candidate, each pixel site has at most seventeen candidates. If any two of the seventeen candidates are the same, only one of them is included in the candidate set. The size of the state space is reduced significantly by this operation.

We take the pixel site $(2i-1, 2j)$ as an example to show the formulation of its candidate set. As shown in Fig. 6.1, centered at the pixel site $(2i-1, 2j)$, a 7×7 window is formed in the high resolution image $H$. The candidates are proposed from pixels in this window. In discrete directions 1, 4, 7, 11 & 14, the two closest neighboring pixels of the center pixel are available and the average of the two

neighboring pixels in the same direction is taken as one candidate pixel. As for the other directions, no closest neighboring pixels are available. In those cases, the pixels at the intersecting points (intersection of the direction and the central $4 \times 2$ square) are interpolated and their average is used as a candidate pixel. For example, direction 8 intersects the central square at points A and B. Pixel values at point A and B are first interpolated as $A = \frac{1}{8}I(2i-3, 2j-1) + \frac{7}{8}I(2i-3, 2j+1)$ and $B = \frac{7}{8}I(2i+1, 2j-1) + \frac{1}{8}I(2i+1, 2j+1)$, and their average is then used as a candidate pixel.

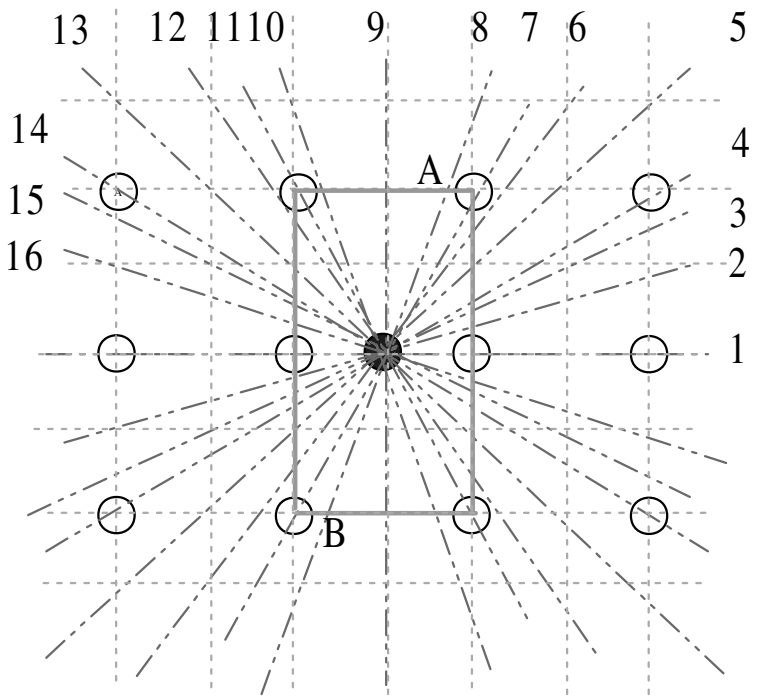



Figure 6.1: Candidate set for pixel site $(2i-1, 2j)$. Circles represent pixels available from the low resolution images.

Proposing the candidates in a 7×7 local area is risky since 7×7 is relatively large. High frequency noise could be introduced easily if the intensity continuity of neighboring pixels is not maintained. However, the sharpness of interpolated edges can benefit from the interpolation from a large local area. In the proposed MRF-EDI method, the possible high frequency noise could be suppressed efficiently by the DAS spatial constraint, whereas the sharpness of interpolated edges is reserved.

The procedures to obtain candidate sets for pixel sites $(2i, 2j-1)$ and $(2i, 2j)$ are similar. Only the central squares have different dimensions. The dimension for pixel site $(2i, 2j-1)$ is $2 \times 4$ and it is $2 \times 2$ for pixel site $(2i, 2j)$.

### 6.3.5 Single-Pass Implementation

Adopting the one-pass algorithm in [89], we designed a single-pass implementation to replace the iterative optimization. Although no iteration is required in the single-pass implementation, its performance is comparable to that of the iterative optimization. The main reason for the single-pass implementation to work well is that the initial state is very close to the global optimal state.

The details of the single-pass implementation are as follows. The low resolution image is initially interpolated using a conventional method, for example, bicubic. Then for each edge pixel, we calculate the single-pixel related energy of each candidate in the candidate set and the one that has the minimal single-pixel related energy is the final output. The single-pixel related energy is defined in (4.5).

## 6.4 Simulation Results

In this section, the "Foreman", "Bicycle" and "Hall" video sequences are used to test the proposed MRF-EDI interpolation algorithm. The performance of the proposed MRF-EDI method is compared to that of the NEDI method [49]. In addition, it is also compared to the traditional bilinear interpolation methods. The implementation of the new edge-directed interpolation algorithm is downloaded from the author's website [90].

The low resolution video sequences are obtained by directly downsampling the original frame images by a factor of two in both row and column dimensions. In the implementation of the proposed MRF-EDI method, the parameters are set as follows. For iterative optimization, the maximal iteration number $K$ is $K = 100$. The original value $T$ is set as $T_0 = 50$ and it is updated according to $T = (.95)^{\lceil \frac{q}{K_s} \rceil} T_0$, where $\lceil x \rceil$ represents the minimal integer that is not smaller than $x$ and $q$ represents the $q$th iteration. $K_s$ denotes the number of iterations that are

run with the same parameter value $T$ and is set as $K_s = 7$. These parameters are set empirically. For single-pass interpolation, $T = 50$ is used for energy calculation and comparison.

Here is a comment on the sensitivity of the algorithm to these parameters. Theoretically, the initial value $T$ must be high and the updating rate must be slow in order to achieve the global minimal state. However, in this practical interpolation problem the initial state of the interpolated image is almost "good" because it's interpolated using well-developed traditional interpolation methods. On the other hand, we don't really care if the final interpolation result is global optimal if only the edges have been reconstructed well. Consequently, the algorithm is not sensitive to these parameters. The same set of parameters is used for various testing sequences and no modification is required.

### 6.4.1 Interpolation Comparison

The results of the proposed MRF-EDI method are compared to those of NEDI and conventional bilinear interpolation methods. Interpolation examples are shown in Fig. 6.2, Fig. 6.4 and Fig. 6.6. The local area zoom-in comparisons are shown in Fig. 6.3, Fig. 6.5 and Fig. 6.7. From Fig. 6.3, one can observe that the major edges interpolated by the proposed MRF-EDI method are sharpest. Fig. 6.5 shows that both MRF-EDI and NEDI methods reconstruct the high-contrast edges very well. However, artificial edges are produced in the texture area by the NEDI method, which makes the texture area look unnatural. This is the problem that is addressed as "losing fidelity to the original image" in [49]. The proposed method is highly locally adaptive and does not have this problem. A texture interpolation example is shown in Fig. 6.8. Artificial edges are produced in NEDI method. The interpolated video sequences in avi format are available at `http://videoprocessing.ucsd.edu/~minli/interpolationdemo.htm`.

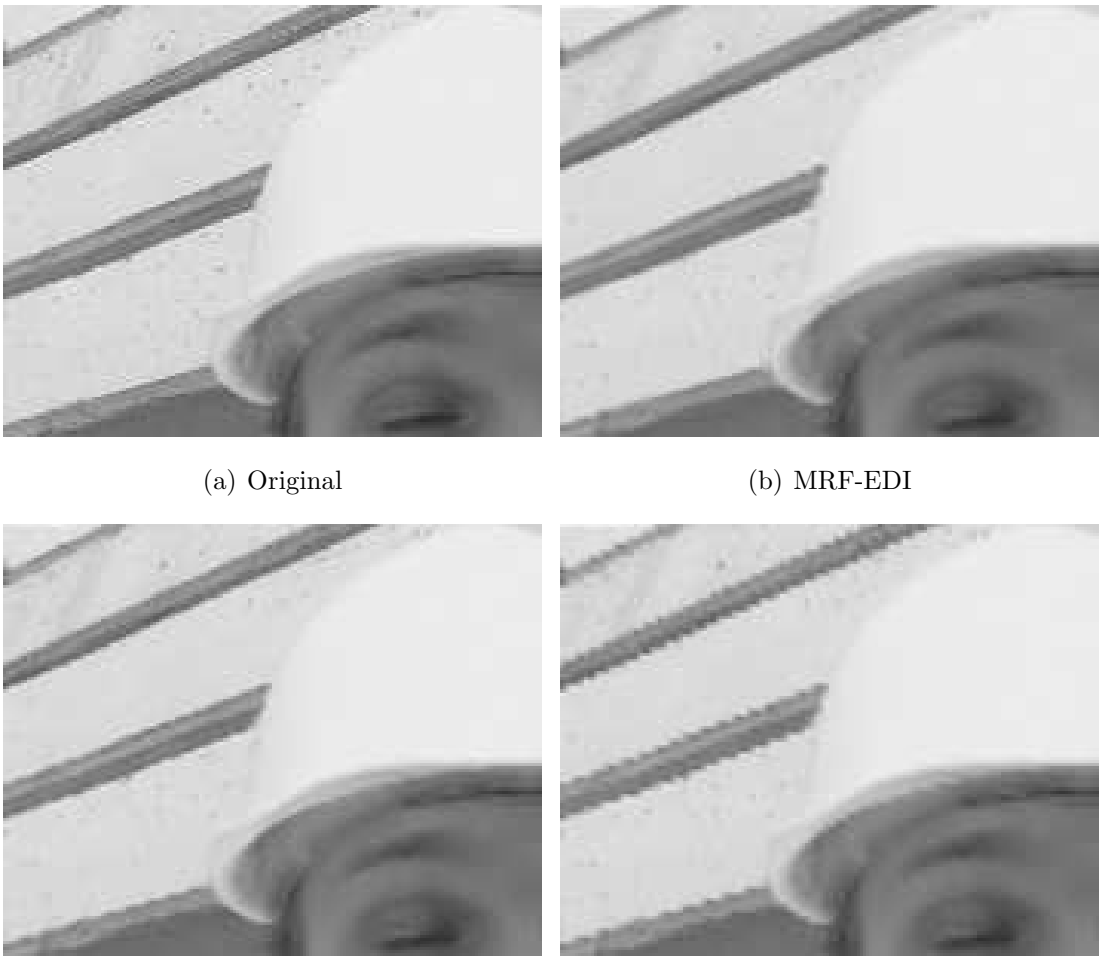

(a) Original

(b) MRF-EDI (33.92 dB)

(c) NEDI (33.50 dB)

(d) Bilinear (32.36 dB)

Figure 6.2: 4× interpolation of “Foreman”.



(a) Original (b) MRF-EDI

(c) NEDI (d) Bilinear

Figure 6.3: Zoom-in comparison.

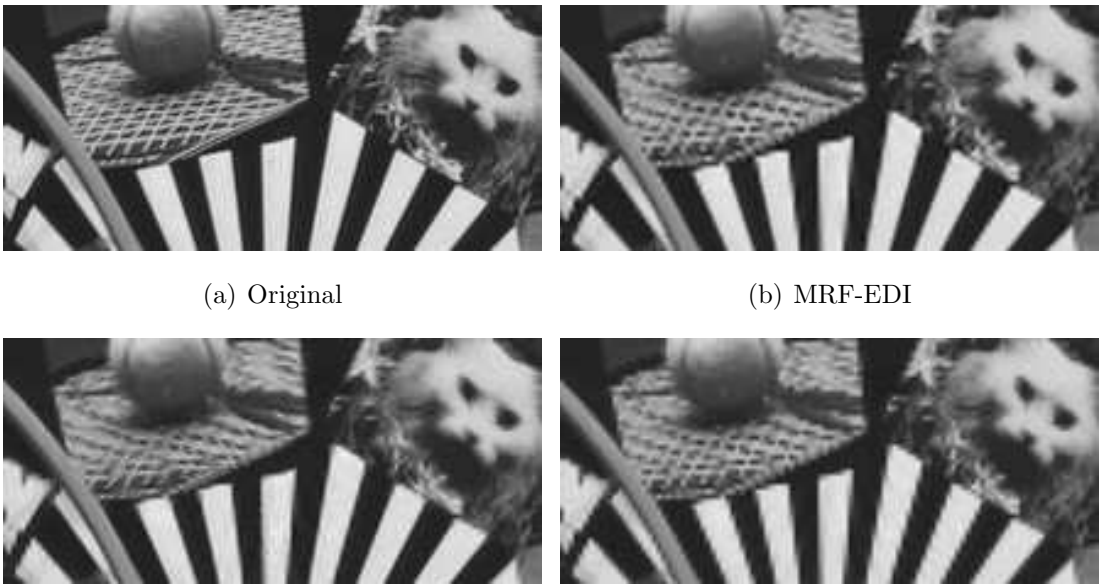
(a) Original

(b) MRF-EDI (29.58 dB)

(c) NEDI (28.75 dB)

(d) Bilinear (28.83 dB)

Figure 6.4: 4× interpolation of “Bicycle”.



(a) Original (b) MRF-EDI

(c) NEDI (d) Bilinear

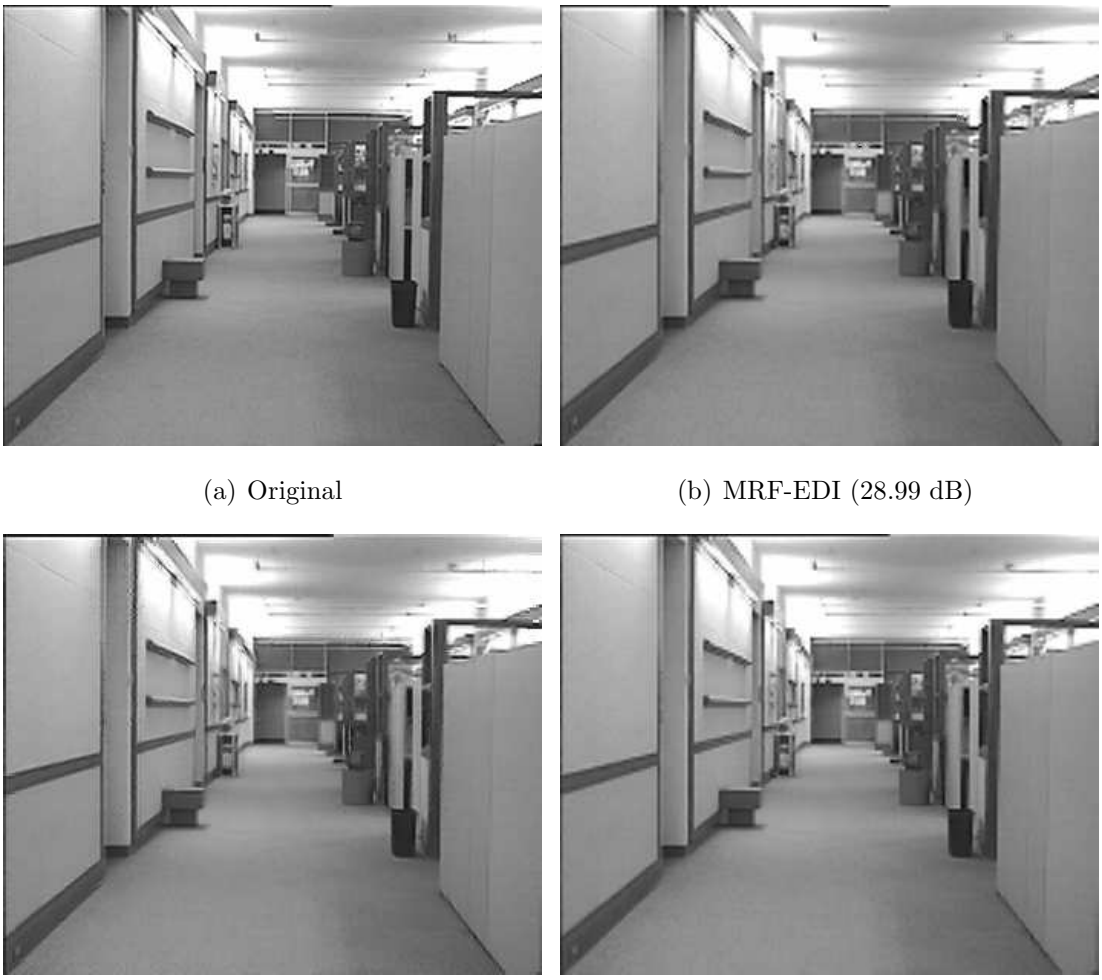
Figure 6.5: Zoom-in comparison.

### 6.4.2 Edge Maps

The proposed MRF-EDI interpolation is only applied to edge pixels in each image. The edge maps of interpolation examples in Subsection 6.4.1 are shown in Fig. 6.9. In the binary map images, bright pixels represent edge pixels. It can be observed that, in the MRF-EDI method, edge and near-edge pixels are identified and are interpolated using the proposed method to achieve strong geometric regularity.

### 6.4.3 Effects of Neighborhood Structure Sizes

For edge directions that are defined by zigzag patterns, a larger neighborhood structure size enables more discrete directions and thus higher resolution of edge directions. Here we show the effects of interpolating the same image with different neighborhood structure. The first frame in the "Bicycle" sequence is used here, of which the original image is shown in Fig. 6.2(a). The maximal neighborhood structure sizes are set as 7×7 and 9×9 respectively. Size 7×7 structure enables sixteen discrete directions while size 9×9 structure enables twenty-four directions. The



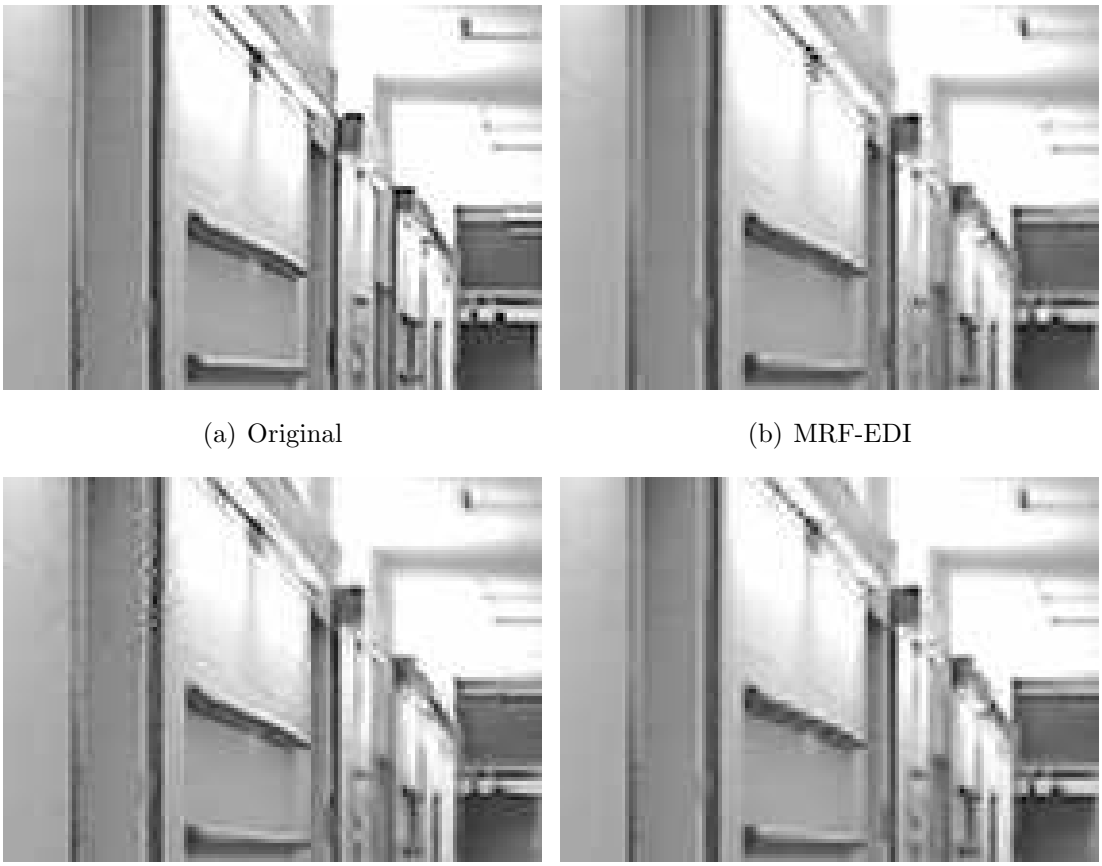
(a) Original

(b) MRF-EDI (28.99 dB)

(c) NEDI (28.26 dB)

(d) Bilinear (28.63 dB)

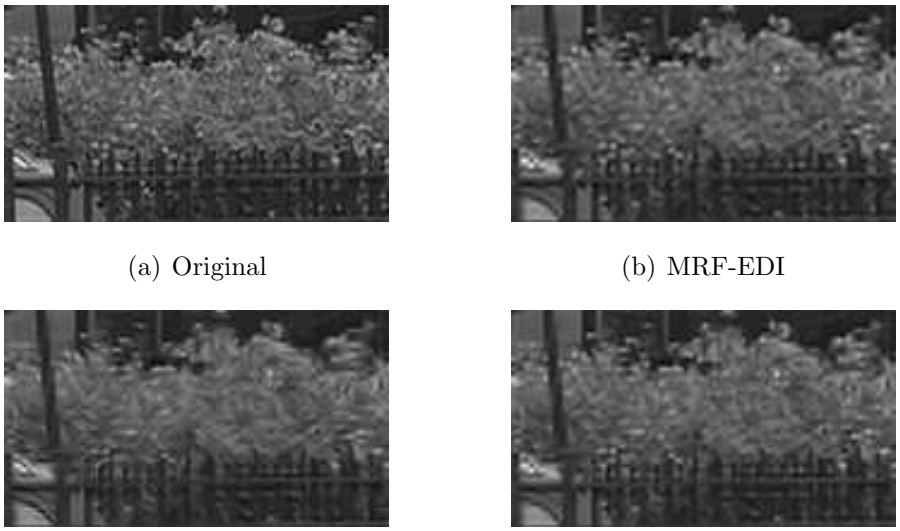
Figure 6.6: 4× interpolation of “Hall”.



(a) Original (b) MRF-EDI

(c) NEDI (d) Bilinear

Figure 6.7: Zoom-in comparison.



(a) Original (b) MRF-EDI

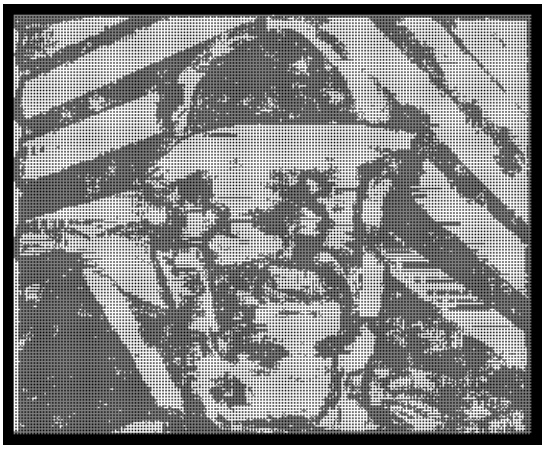
(c) NEDI (d) Bilinear

Figure 6.8: Texture interpolation comparison.

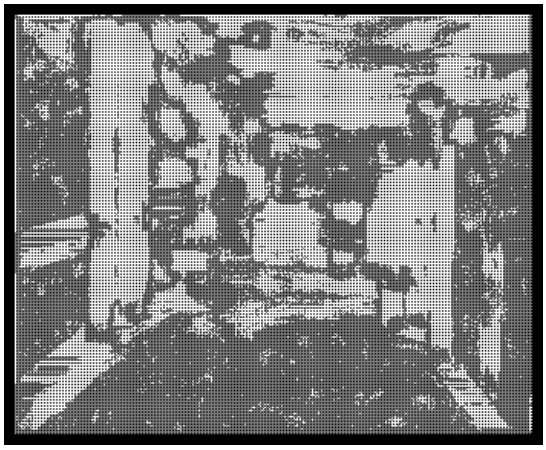
data window size for weights calculation for sixteen directions is 11×11 and 15×15 for twenty-four directions. The interpolated frames are shown in Fig. 6.10(a) and 6.10(b). There is a minor difference in PSNR levels. Subjectively, from the zoom-in region comparisons shown in Fig. 6.10(c) and 6.10(d), one near-horizontal edge has been reconstructed more smoothly (along-edge direction) in size 9×9 structure case. Basically we can conclude that higher edge direction resolution leads to more exact interpolation of edges in arbitrary directions. Certainly, the computational complexity would increase accordingly.

## 6.5 Conclusions

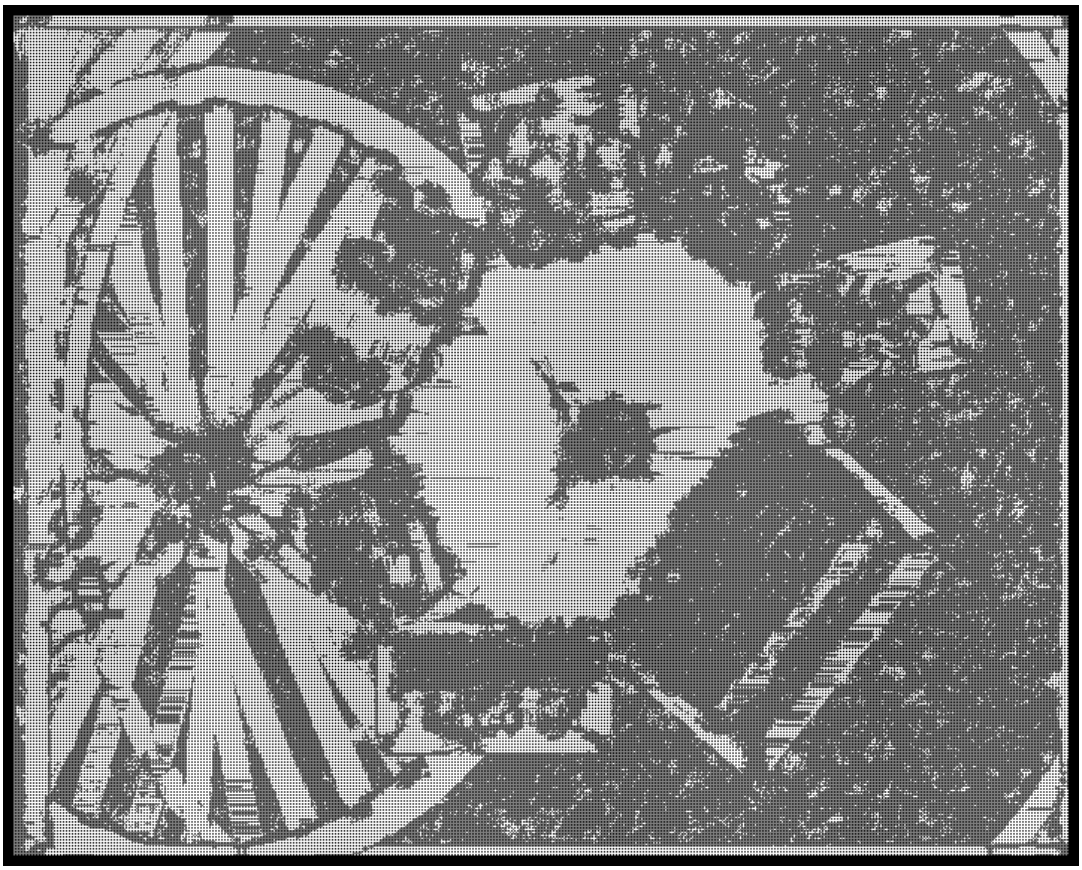
An implicit edge-directed interpolation algorithm for natural images is proposed in this chapter. It is an MRF model-based edge-directed interpolation method. In this method, the interpolated image is modeled as an MRF and the most desired interpolated image is related to the minimal energy state of a two-dimensional random field. Edge direction information is incorporated when formulating the energy function. Consequently, energy that is along edge directions is strongly suppressed to achieve smoothness while energy that is across edge directions is much less sup-



(a) “Foreman”

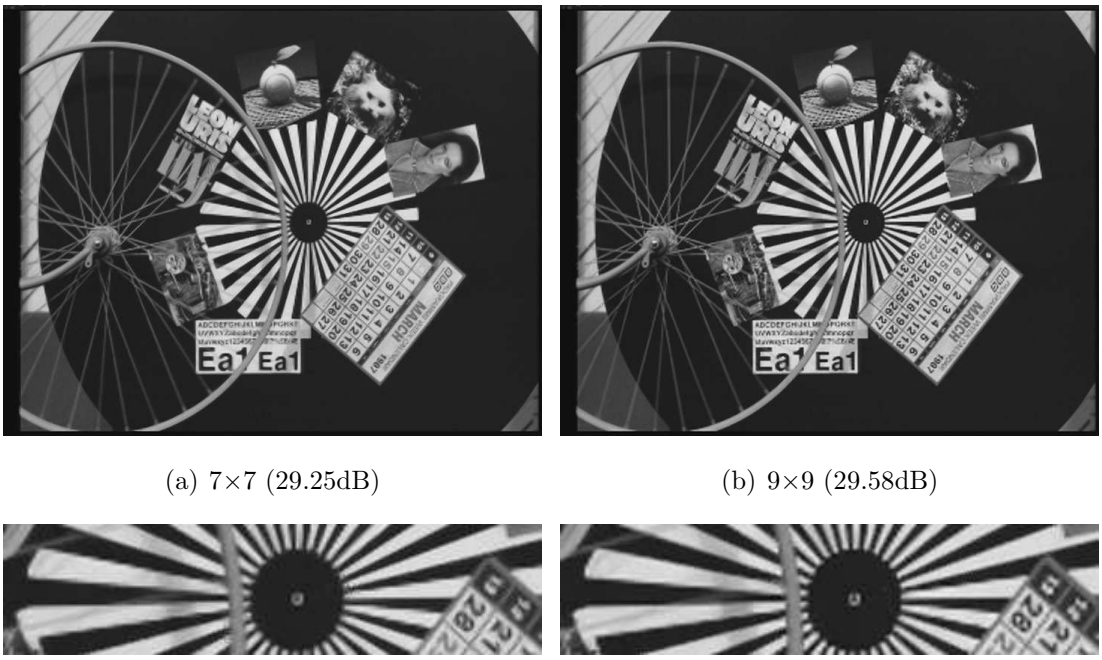


(b) “Hall”



(c) “Bicycle”

Figure 6.9: Edge pixel maps.



(a) 7×7 (29.25dB) (b) 9×9 (29.58dB)

(c) 7×7 (d) 9×9

Figure 6.10: Performance comparison of different neighborhood structure sizes.

pressed to maintain the sharpness of edges. As a result, the interpolated edges have strong geometric regularity.

The edge direction information has been obtained using a statistical-based approach, which should be more robust than discrete differentiation methods. Consequently, the edge direction information of each edge pixel is represented by a weight vector, which is integrated into the formulation of the energy function in the GMRF model.

To lower the complexity, the proposed MRF-EDI method is only applied to edge pixels. To discriminate edge pixels from others, consistency in edge directions is checked, which helps avoid the interpolation of spurious minor edges in texture areas. Simulation results show that the proposed MRF-EDI method produces interpolated major edges with strong geometric regularity. Comparing to bilinear interpolation, the proposed method improved the visual quality of the interpolated edges while maintaining high PSNR. Compared to the NEDI method, the proposed MRF-EDI method provides sharper edges with higher PSNR.

The text of this chapter is adapted from material that has been submitted for publication as: M. Li and T. Q Nguyen, *Markov Random Field Model-Based Edge-Directed Image Interpolation*, *IEEE Transaction on Image Processing* and also appears in part as: M. Li and T. Q Nguyen, *Markov Random Field Model-Based Edge-Directed Image Interpolation*, *IEEE International Conference on Image Processing* 2007. . The dissertation author was the primary researcher of this publication, and the co-author listed directed and supervised the research which forms the basis for this chapter.

# 7

# Scalable Video Coding

Video broadcasting has a wide range of applications, such as Internet video, wireless LAN video, mobile wireless video for conversational and live broadcasting purposes, multi-channel video production and distribution, and layered protection of contents. Scalable video codecs [91, 92] have gathered much attention due to flexibilities they offer in terms of spatial, temporal, signal-to-noise ration (SNR), and complexity scalabilities [93]. Most SVC techniques can be classified as Wavelet-based SVC [94, 95] or H.264/AVC-based SVC [96].

As an important part of scalable video codecs, low complexity and low bitrate encoders have important applications. Firstly, the video bitstream can be sent to different devices which most likely vary in terms of the complexity levels and power characteristics. The received video bitstream should therefore adapt accordingly. Secondly, the characteristics of the transmission channels and of the receiving devices are unknown at the beginning of the transmission. In this case, the transmission should start with a low bit rate bitstream and can later switch to higher bit rates if the channel and receiving devices' profiles allow. Lastly, the characteristics of the transmission channels and of the receiving device's can change dynamically. For example, the device should be able to trade off the received video quality for longer power life. A low-complexity and low bit rate scalable video coding scheme is necessary for these applications.

## 7.1 Optimal Wavelet Filter Design in Wavelet-Based SVC

### 7.1.1 Introduction

Although the current reference codec (H.264-based SVC) for SVC is not wavelet-based, it is still a very active research area because the wavelet transform naturally enables spatial scalability. Consequently, many researchers are exploring applications of wavelet theories in SVC [97].

The spatial scalability in both wavelet-based SVC and H.264/AVC-based SVC is achieved by using lowpass filtering followed by downsampling. The downsampling filter $H_0(z)$ in H.264/AVC-based SVC can be any good lowpass filter, e.g., the MPEG filter [98] or the JPEG filter [99]. In contrast, filter $H_0(z)$ has to be a good wavelet lowpass filter with a smooth basis function in wavelet-based SVC [95].

The Low Band Correction (LBC) technique is proposed in [95] to improve the spatial scalability. As indicated in Fig. 7.1, in LBC, the prediction error frames,

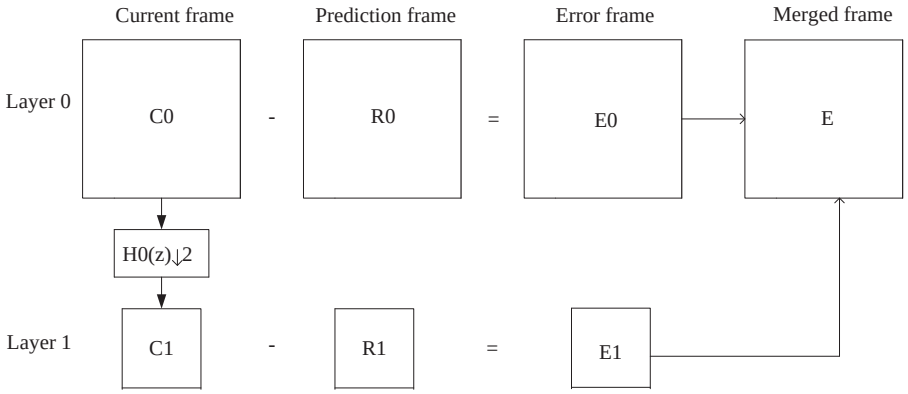



Figure 7.1: LBC concept.

$E_0$ and $E_1$, from different spatial layers, layer 0 and layer 1, can be merged into one texture frame $E$. This merge effectively reduces the overhead texture information in SVC. Moreover, at the decoder side, prediction error frames $E_0$ and $E_1$ can be derived from merged texture frame $E$ under certain conditions. In the LBC

technique, the merged texture frame $E$ is generated according to

$$E = E_0 - F_0(z)(\uparrow 2)(\downarrow 2)H_0(z)\{E_0\} + F_0(z)(\uparrow 2)\{E_1\}. \tag{7.1}$$

where filters $F_0(z)$ and $H_0(z)$ are the upsampling and downsampling filters, respectively. If filters $F_0(z)$ and $H_0(z)$ satisfy the condition

$$(\downarrow 2)H_0(z)F_0(z)(\uparrow 2)\{X\} = X, \tag{7.2}$$

where $X$ represents an arbitrary signal, the prediction error frames $E_1$ and $E_0$ can be derived from the merged texture frame $E$ as

$$E_1 = (\downarrow 2)H_0(z)\{E\} \tag{7.3}$$

and

$$E_0 = E - F_0(z)(\uparrow 2)\{E_1\} + F_0(z)(\uparrow 2)(\downarrow 2)H_0(z)\{E_0\}, \tag{7.4}$$

where

$$(\downarrow 2)H_0(z)\{E_0\} = C_1 - (\downarrow 2)H_0(z)\{R_0\}. \tag{7.5}$$

The condition in (7.2) is actually a half-band condition, which can be expressed as

$$P(z) + P(-z) = z^{-L}, \tag{7.6}$$

where $P(z) = H_0(z)F_0(z)$ is a halfband filter and $L$ is the delay parameter. Consequently, the downsampling filter $H_0(z)$ in wavelet-based SVC [95] has to be a 'good' wavelet filter. For a wavelet filter to be 'good', the following properties are important.

- The lowpass filter of the analysis filter bank, $H_0(z)$, and the lowpass filter of the synthesis filter bank, $F_0(z)$, have to satisfy the halfband condition: $P(z) = H_0(z)F_0(z)$ is a halfband filter.

- The magnitude response of the lowpass filter $H_0(z)$ should have flat passband, sharp transition between passband and stopband, and large stopband attenuation. The requirements on the magnitude response of the lowpass filter $F_0(z)$ are not as strict as those of the filter $H_0(z)$, however, $F_0(z)$ should still have acceptable lowpass magnitude response.

- The filters $H_0(z)$ and $F_0(z)$ should have a certain regularity order to guarantee the smoothness of the corresponding scaling functions and wavelets. This property is important for image filtering and reconstruction applications.

According to the requirements stated above, it is not difficult to verify that the MPEG downsampling filter is not a good wavelet filter. Let $H_0(z)$ be the MPEG filter and design the filter $F_0(z)$ to satisfy the half band condition in (7.6). The highpass filters $H_1(z)$ and $F_1(z)$ are chosen to satisfy the aliasing cancellation condition [100], $H_1(z) = -F_0(-z)$ and $F_1(z) = H_0(-z)$. The magnitude responses of filters $H_0(z)$ and $H_1(z)$ are shown in Fig. 7.2(b). $H_1(z)$ has a large "bump" in the stopband which limits the corresponding wavelet system's application for video compression. Thus, the MPEG filter does not work in the proposed scheme and the Daubechies (9,7) filters are used in [95]. The magnitude response of Daubechies length-9 filter is shown in Fig. 7.2(a). The main problem of using Daubechies (9,7) filters is that the downsampling filter $H_0(z)$ has too much energy leakage in the stopband which significantly degrades the full search motion estimation performance at lower spatial resolution. This point is shown using simulation results in Section 7.1.4.

In our design, we formulate the design problem as the design of a class of wavelet filters, which are optimal in the sense that the best tradeoff between high stopband attenuation of filter $H_0(z)$ and flat passband response of filter $F_0(z)$ is achieved.

The organization of this section is as follows. The design procedure is presented in Sections 7.1.2 and 7.1.3. Some design examples are shown in Section 7.1.4 along with the application to real video sequences. The performance of the designed filters is compared to that of the Daubechies (9,7) filters [101] and the MPEG filter.

### 7.1.2 Design Procedure

The Daubechies length-9 filter is used as a prototype lowpass filter $H_0(z)$. Two of four zeros at $\pi$ of the prototype filter are retained while the other two zeros are moved along the unit circle towards $\pi/2$ and $-\pi/2$ to obtain other solutions.

Both the frequency selectivity property and the smoothness of the scaling filter are essential for the compression performance of the wavelet system, which yields the following cost function: $\Phi = \alpha_0$(stopband energy of $H_0(e^{j\omega})$)+$\alpha_1$(1/Hölder regularity [102] of $H_0(z)$)+$(1-\alpha_0-\alpha_1)$(1/Hölder regularity of $F_0(z)$) over the pair of moving zeros of $H_0(z)$.

The pair of moving zeros of $H_0(z)$ can be specified by a single angle $\theta$ so the design problem is to minimize $\Phi$ over the angle $\theta$. A series of angle $\theta$s are specified and the one that minimizes $\Phi$ yields the optimal solution $h_0(n)$ and $f_0(n)$. Lowpass filters $h_0(n)$ and $f_0(n)$ are related via the halfband condition.

The following parameters have to be specified at the beginning of the optimization process.

- the stopband frequency, $\omega_s$, of the lowpass filter $H_0(z)$.
- the weighting factors $\alpha_0$ and $\alpha_1$, which enable tradeoff between frequency selectivity and smoothness of the corresponding scaling functions and wavelets, $0 < \alpha_0, \alpha_1 < 1$, and $0 < \alpha_0 + \alpha_1 < 1$.
- the regularity order of filter $F_0(z)$. To guarantee its smoothness, the design specifies its regularity order, i.e., number of zeros at $\pi$. For simplicity, its regularity order is even in the current design program.

The program can be modified to design longer length filters by adding one or more pairs of zeros to $H_0(z)$.

### 7.1.3 Halfband Condition Specification Using Matrix Formulation

Given $H_0(z)$, this section presents a method to design the corresponding filter $F_0(z)$ such that they form a perfect reconstruction system. Keep in mind that the highpass filter $H_1(z)$ and $F_1(z)$ are obtained by the aliasing cancellation condition. It implies that the convolution of $h_0(n)$ and $f_0(n)$, i.e., $p(n) = h_0(n) * f_0(n)$, is a halfband filter. The following derivation assumes that all filters have zero phase and odd length though a similar derivation can be done for other cases.

The halfband constraint can be expressed as

$$p(n) = \sum_{k=-\frac{(L-1)}{2}}^{\frac{(L-1)}{2}} h_0(n-k) f_0(k), \tag{7.7}$$

where $p(n)$ is a halfband filter and L is the length of filter $f_0(n)$. The coefficients of $p(n)$ have the property

$$p(n) = \begin{cases} 0.5 & n = 0 \\ 0 & n = \pm 2, \pm 4, \cdots \\ p(-n) & n = 1, 3, \cdots \end{cases}. \tag{7.8}$$

For symmetric filters $f_0(n)$, the convolution in (7.7) can be rewritten linearly [103] as

$$p(n) = \sum_{k=1}^{\frac{(L-1)}{2}} (h_0(n-k) + h_0(n+k)) f_0(k) + h_0(n) f_0(0). \tag{7.9}$$

Eq. (7.10) shows the matrix formulation for the case where the lengths of $h_0(n)$ and $f_0(n)$ are 9 and 11, respectively. Correspondingly, filter $p(n)$ has length 19.

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.5 \end{bmatrix} = \underbrace{\begin{bmatrix} p(8) \\ p(6) \\ p(4) \\ p(2) \\ p(0) \end{bmatrix}}_{\mathbf{P}}$$

$$= \underbrace{\begin{bmatrix} h_0(3) & h_0(4) & 0 & 0 & 0 & 0 \\ h_0(1) & h_0(2) & h_0(3) & h_0(4) & 0 & 0 \\ h_0(-1) & h_0(0) & h_0(1) & h_0(2) & h_0(3) & h_0(4) \\ h_0(-3) & h_0(-2) & h_0(-1) & h_0(0)+h_0(4) & h_0(1)+h_0(3) & h_0(2) \\ 0 & 2h_0(-4) & 2h_0(-3) & 2h_0(-2) & 2h_0(-1) & h_0(0) \end{bmatrix}}_{\mathbf{convh}} \begin{bmatrix} f_0(5) \\ f_0(4) \\ f_0(3) \\ f_0(2) \\ f_0(1) \\ f_0(0) \end{bmatrix}. \tag{7.10}$$

In addition to the halfband condition, the filter $f_0(n)$ has to have a certain regularity order in order to yield smooth basis functions. In our design, $f_0(n)$ takes the following form:

$$F_0(z) = z^{\frac{R}{2}} (1 + z^{-1})^R F'(z), \tag{7.11}$$

where R is even. If R is odd, the first term becomes $z^{\frac{R-1}{2}}$.

Consequently, (7.11) can be formulated in matrix form. An example for R=2 is

$$\begin{bmatrix} f_0(5) \\ f_0(4) \\ f_0(3) \\ f_0(2) \\ f_0(1) \\ f_0(0) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 2 & 2 \end{bmatrix}}_{\mathbf{smth}} \underbrace{\begin{bmatrix} f'(4) \\ f'(3) \\ f'(2) \\ f'(1) \\ f'(0) \end{bmatrix}}_{\mathbf{f'}}. \tag{7.12}$$

Considering (7.10) and (7.12) together, we can design the filter $f'$ by choosing the last column of matrix $[(\mathbf{convh})(\mathbf{smth})]^{-1}$ and scaling it by $\frac{1}{2}$. The corresponding designed lowpass filter $f_0(n)$ can be obtained from $f'$ according to (7.11).

The length of the designed filter $f_0(n)$ is limited as follows. The matrix $[(\mathbf{convh})(\mathbf{smth})]$ has to be square to allow the existence of the inverse matrix. Suppose the halfband filter p is of length (4K+3) for some integer K. Consequently, the vector $\mathbf{p}$ in (7.10) is of length K+1, and thus the row dimension of matrix (**convh**) has to be K+1. To guarantee that the product matrix $[(\mathbf{convh})(\mathbf{smth})]$ is square, the column dimension of matrix (**smth**) must be K+1. Thus, the length of vector $\mathbf{f'}$ in (7.12), which is (L-R-1)/2+1 for a length-L, regularity order-R filter $f_0(n)$, equals K+1. Consequently we have

$$K + 1 = (L - R - 1)/2 + 1. \tag{7.13}$$

Suppose filter $h_0(n)$ is of length N. Another obvious equation is

$$N + L - 1 = 4K + 3, \tag{7.14}$$

which comes from the fact that the convolution of $h_0(n)$ and $f_0(n)$ is the halfband filter $p(n)$. From (7.13) and (7.14), we can obtain L=N+2R-2. This result is valid only for even Rs. Eq. (7.13) becomes

$$K + 1 = (L - R)/2 \tag{7.15}$$

for odd Rs and the length of $f_0(n)$ becomes N+2R.

In summary, the length of $f_0(n)$ is

$$L = \begin{cases} N + 2R - 2 & \text{even R} \\ N + 2R & \text{odd R} \end{cases} . \tag{7.16}$$

### 7.1.4 Simulation Results

Four sets of filters, New1, New2, New3 and New4 are designed using the proposed method. Their magnitude responses and impulse responses of their scaling functions are shown in Fig. 7.2.

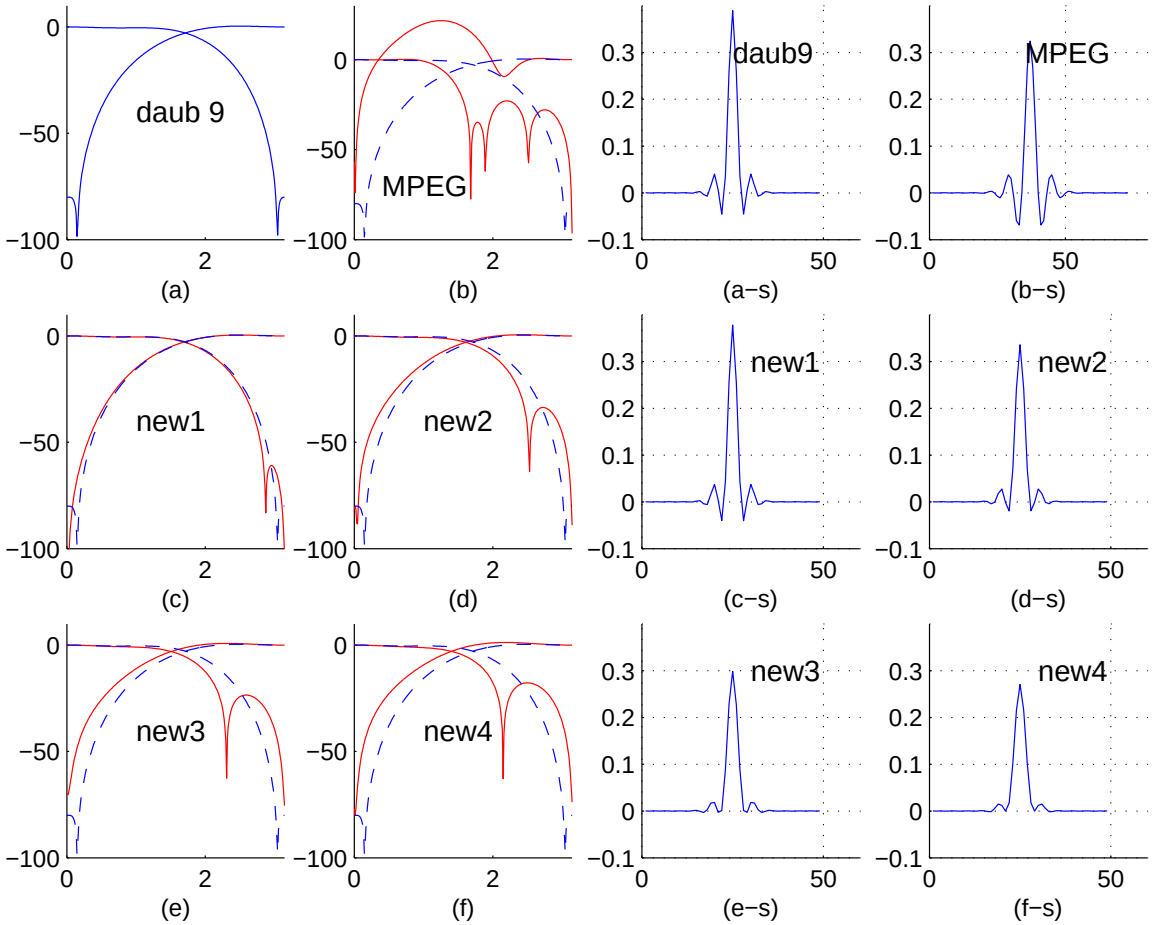



Figure 7.2: Design examples. (a), (b), (c), (d), (e) and (f) The magnitude responses of the filters. (a-s), (b-s), (c-s), (d-s), (e-s) and (f-s) the scaling functions corresponding to $h_0(n)$.

The tradeoff between high stopband attenuation of the analysis lowpass filter

Table 7.1: Average PSNRs (dB) of prediction frames

| | | Crew | City | Soccer | Harbour |
|---|---|---|---|---|---|
| MPEG filter | layer1 | 30.6 | 29.5 | 28.2 | 25.0 |
| | layer2 | 31.0 | 30.9 | 27.1 | 27.2 |
| New3 filter | layer1 | 30.2 | 27.7 | 27.8 | 24.6 |
| | layer2 | 30.6 | 29.8 | 27.0 | 26.4 |
| Daubechies L-9 filter | layer1 | 29.8 | 26.4 | 27.3 | 24.2 |
| | layer2 | 29.9 | 28.4 | 26.4 | 25.2 |

$H_0(z)$ and flat passband response of the synthesis lowpass filter $F_0(z)$ can be clearly seen from plots in Fig. 7.2.

The spatial scalability performance of New3 filter set is compared to that of the Daubechies (9,7) filters and MPEG filter. The 4CIF video sequences City, Harbour, Crew and Soccer are filtered and downsampled frame by frame to obtain the CIF resolution video sequences using the MPEG filter, New3 lowpass filter and Daubechies length-9 filter. Full search motion estimation is performed on each adjacent two frames in the CIF sequences. The average PSNRs of the obtained prediction frames are listed in Table 7.1.

The standard CIF video sequences are applied when calculating PSNRs. The QCIF sequences that are obtained via downsampling of CIF video sequences are applied when calculating the PSNRs of QCIF prediction frames, since the standard QCIF sequences are not available. From the table, we can see that the New3 filter set out-performs Daubechies (9,7) filters in performing spatial scalability. Please see the website `http://videoprocessing.ucsd.edu/demo.htm` for the PSNR plots of each predicted video sequence. In addition, the wavelet system built based on the New3 filter has good performance for compression applications.

### 7.1.5 Conclusion

Using the maximal flat wavelet filters as prototype filters, we develop the procedure to design a class of wavelet filters, suitable for both motion estimation and compression in SVC. This method is applicable to any halfband filter, not necessarily to be maximally flat. From the magnitude response, the Daubechies (9,7)

filters have a flat passband, but the length-9 lowpass filter has large energy leakage in the stopband. On the other hand, the MPEG filter has high stopband attenuation and sharp transition band, but the corresponding lowpass synthesis filter has a big 'bump' in its passband which limits its application for video compression (see Fig. 7.2b).

The proposed design method enables a tradeoff between the features of Daubechies (9,7) filters and those of the MPEG filter.

The design method incorporates all of the wavelet filter design requirements and is very efficient. Several parameters are adapted in the design which enables flexibility. The parameters can be adjusted according to particular application requirements.

## 7.2 Motion Vector Prediction in Low-Complexity Low Bit-Rate H.264-Based SVC

### 7.2.1 Introduction

In the H.264-based scalable video model, a low complexity implementation has not been dealt with as an independent issue. However, it can be accomplished to some extent by exploiting several aspects. The scalable video coding model [104] has a choice between a fast motion estimation algorithm [105] and full search motion estimation algorithm. A faster motion estimation method can lower the encoder complexity considerably. Another way to achieve low complexity is to transmit the base layer bitstream initially with none or fewer enhancement layers sent afterwards.

In the current scalable video model, motion vectors have to be coded and sent for all the spatial layers even under low complexity and low bit rate requirements. Otherwise, the full resolution video sequence cannot be reconstructed at the decoder side. Such complexity for motion estimation and rate allocation for motion vectors may not satisfy those applications that demand even lower complexities and lower bit rates. In order to avoid complexity and high motion bit rates, efficient

inter-layer motion vector interpolation techniques are required.

We propose a new inter-layer motion vector interpolation method for low complexity, low bit-rate applications. The proposed scheme consists of a novel weighted smoothing method and a novel scheme called interpolation mode map to efficiently interpolate motion vectors between layers. The motion estimation is performed at the lowest resolution and it does not consume much system resources since the lowest resolution layer normally has very few macroblocks. The motion vector fields for all other spatial layers are interpolated from that of the lowest spatial resolution layer and no other motion estimation is required. The interpolation for higher layers is performed by the mode map method which chooses the best interpolation method for a particular block. Consequently, no motion vectors except for those in the lowest spatial resolution layer are needed to be coded and transmitted, thus both the encoder complexity and the transmission bit rate decrease significantly.

The organization of this section is as follows. The combined interpolation scheme and the mode map method that is used to record the chosen interpolation scheme for a particular block are proposed in Subsection 7.2.2. Four separate inter-layer motion vector interpolation methods are presented in Subsection 7.2.3. Simulation results and discussions are presented in Subsection 7.2.4, which is followed by conclusions in Subsection 7.2.5.

### 7.2.2 Mode map method

In this section, we propose the mode map method as a part of a combined motion vector interpolation scheme. The proposed low-complexity scalable video coding scheme combines four inter-layer motion vector interpolation methods namely: repeat, smoothing, weighted smoothing 1 and weighted smoothing 2, for the interpolation of motion vectors at the higher spatial layers. The concepts and detailed operations of each of the four motion vector interpolation methods are presented in Section 7.2.3. For each spatial layer, a mode map is generated to indicate the chosen interpolation method for a particular block. One example mode map is shown in Fig. 7.3. This map has only four different values, 0, 1, 2 and 3. Thus it has a small entropy and can be coded very efficiently. The mode map method is

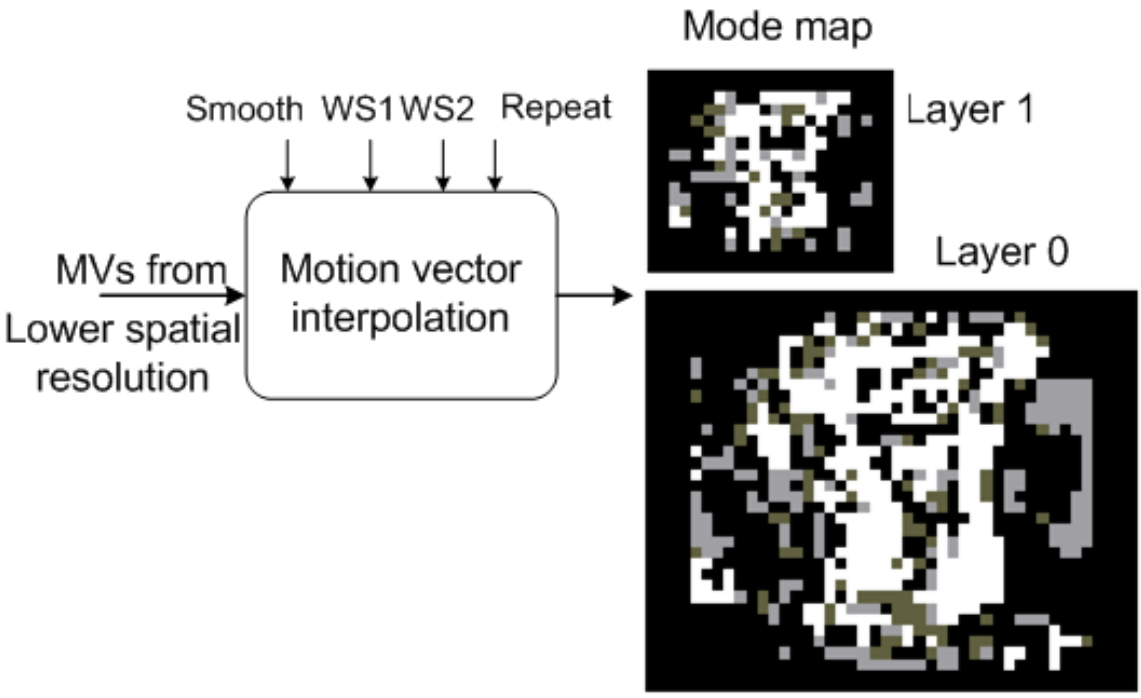



Figure 7.3: The block diagram and an example of interpolation mode map.

much more efficient compared to coding the interpolated motion vectors directly. Each macroblock has two motion vector components and the values of both components can vary over a wide range. As a result, the interpolated motion vector field has higher entropy than its mode map. The tradeoff for the improved coding efficiency is that the corresponding interpolation method has to run at the decoder side as well. If too much computation is involved in the interpolation algorithm, the decoder complexity will increase significantly. Fortunately, the motion vector interpolation methods in the proposed scheme are of very low complexity.

### 7.2.3 Combined Motion Vector Interpolation Method

As mentioned in Section 7.2.2, four motion vector interpolation methods, repeat, smoothing, weighted smoothing 1 and weighted smoothing 2 are involved in the combined method. In this section, we present the four methods one by one while focusing on the weighted smoothing motion vector interpolation methods.

### Repeat method

The repeat method is the most direct inter-layer interpolation method. The motion vector of a particular block at the lower resolution layer is scaled by a factor of 2 and becomes the motion vectors of the four blocks that are at the same positions at the higher resolution layer. The main problem with this method is that the motion compensated frames that are obtained according to the interpolated motion vector information suffer from significant blocking artifacts.

### Smoothing method

A smooth motion vector interpolation technique is proposed in [106]. In this technique, the local smoothness measurement of a motion vector field is defined as

$$\Psi = \Psi_N + \Psi_S + \Psi_E + \Psi_W + \Psi_D + \Psi_C, \tag{7.17}$$

where each term corresponds to differences between various pairs of adjacent vectors in various directions such as north, south, east, west, diagonal and center. To obtain maximally smooth motion vectors, the cost function in (7.17) is minimized. The minimization of $\Psi$ in (7.17) is achieved by setting its first order differentiation to be zero since it is convex.

The x and y components of the motion vectors are processed independently in this scheme. Although the resulting smooth motion vector field reduces blocking, undesirable smoothness at motion boundaries is introduced and could degrade the visual quality at the motion boundaries.

### Weighted smoothing methods

In situations where the adjacent motion vectors that are being smoothed lie within the body of a moving object, minimizing the difference between them is natural. In cases where a block is on an object boundary, it is possible that the motion vectors of neighboring blocks point in different directions. Smoothing such vectors would result in undesirable artifacts [107, 108]. Hence, the optimization problem for obtaining smooth motion vectors would perform better if the terms in

(7.17) being minimized are weighted by the correlation between the motion vectors involved in that term.

Two weighted smooth inter-layer motion vector interpolation methods are described as follows. The cost function in the first Weighted Smoothing approach, $WS_1$, is

$$\Psi = w_X\Psi_X + w_Y\Psi_Y + w_{D_1}\Psi_{D_1} + w_{D_2}\Psi_{D_2} + \Psi_C, \tag{7.18}$$

where the $_X$, $_Y$, $_{D_1}$ and $_{D_2}$ directions are indicated by arrows in Fig. 7.4(a). It can be seen as a weighted expansion of the object function in (7.17).

In the second Weighted Smoothing approach, $WS_2$, the cost function formulation follows similar principles as in approach $WS_1$. Only the directions are formed differently as shown in Fig. 7.4(b). The cost function is

$$\begin{aligned}\Psi &= w_N\Psi_N + w_S\Psi_S + w_W\Psi_W + w_E\Psi_E + w_{D_1}\Psi_{D_1} \\ &\quad + w_{D_2}\Psi_{D_2} + w_{D_3}\Psi_{D_3} + w_{D_4}\Psi_{D_4} + \Psi_C,\end{aligned} \tag{7.19}$$

In the following, we present the expressions for the $\Psi_i$s in (7.18) and (7.19), where i=$N, S, W, E, D_1, D_2, D_3, D_4, C$, and show how the weights for $\Psi_i$s are computed. After weights are calculated, the differential method proposed in [106] can be used to find the solution of (7.18) and (7.19) such that the weighted smoothness is maximized. The convexity of objective function (7.17) is preserved when it is expanded to the weighted forms in (7.18) and (7.19).

### $\Psi_i$ calculation

Referring to Fig. 7.4(a), the expressions for $\Psi_Y, \Psi_{D_1}$ and $\Psi_C$ in (7.18) can be written as

$$\begin{aligned}\Psi_Y &= (V_N - V_1)^2 + (V_1 - V_3)^2 + (V_3 - V_S)^2 + (V_N - V_2)^2 \\ &\quad + (V_2 - V_4)^2 + (V_4 - V_S)^2\end{aligned} \tag{7.20}$$

$$\Psi_{D_1} = (V_{d1} - V_1)^2 + (V_1 - V_4)^2 + (V_4 - V_{d4})^2 \tag{7.21}$$

$$\Psi_C = (V_1 - V_C)^2 + (V_2 - V_C)^2 + (V_3 - V_C)^2 + (V_4 - V_C)^2 \tag{7.22}$$

The $X(i.e., horizontal)$ and $D_2$ directions can be written in a similar fashion to (7.20) and (7.21) respectively. In these equations, $V_1, V_2, V_3$ and $V_4$ are unknown motion vectors and other $V_i$s are neighboring motion vectors at the lower resolution layer as shown in the Fig. 7.4(a).

**Weight calculation**

The proposed weighted smoothing methods utilize correlation between the motion vectors for improved motion vector interpolation [107, 108]. The correlation information is used to find weights in different ways based on how the vectors are grouped into $\Psi_i$. Two variants are proposed here, one corresponds to approach $WS_1$ and the other corresponds to approach $WS_2$.

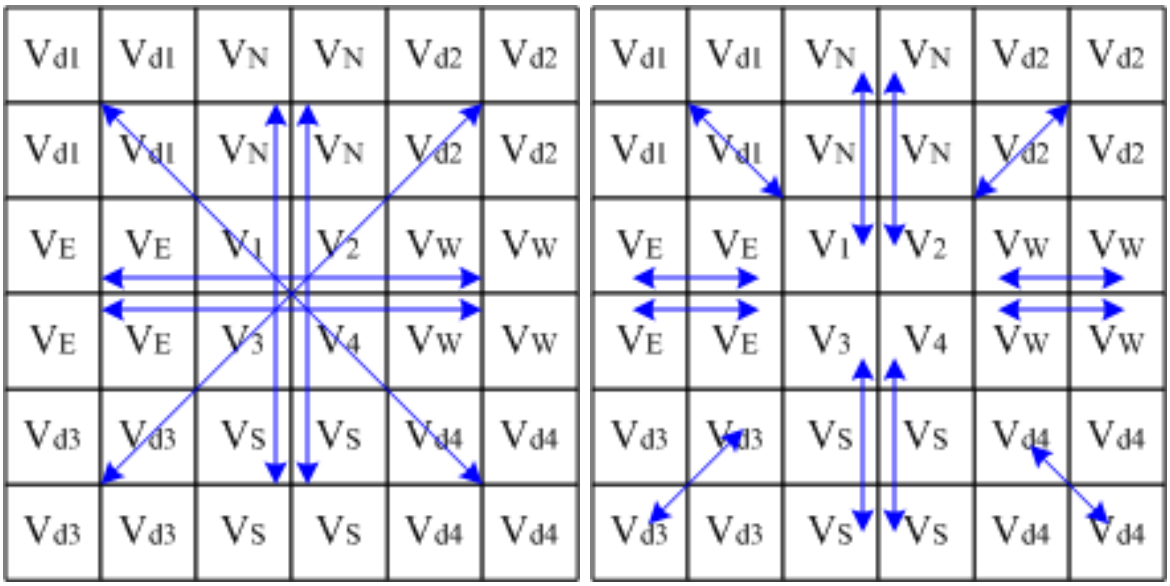
In the $WS_1$ approach, the relative orientation of the vectors along each of these axes in the base layer motion vector field determines the weights $w_i$. The normalized dot product is computed between each of the three pairs of vectors along each direction in the lower resolution image. Let $a_1$, $a_2$, $a_3$ refer to the normalized dot products for a particular direction, say horizontal. Then the weight $w_X = w^2$ in that direction is computed as

$$w = \begin{cases} \frac{a_1+a_2+a_3}{3} & a_1 + a_2 + a_3 > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{7.23}$$

If the sum of the normalized dot product is negative, the vectors along this axis are completely uncorrelated and should not be smoothed. It is clear that if all three vectors along an axis are in the same direction, the weight is 1 for that axis. Also, the center has a default weight of 1. Hence smoothness at the center, which involves minimizing differences between each of $V_1, V_2, V_3, V_4$ and $V_c$ has a higher weight.

In the $WS_2$ approach, the correlation between $V_C$ and each of its 8-neighbors at the lower spatial resolution layer given by the normalized dot product is directly used as a weight to minimize the corresponding term in the cost function, i.e., $w_N$ is obtained by the dot product of $V_N$ with $V_C$.

We note that when one of the motion vectors is zero and the other motion vector's magnitude is zero or close to zero, the correlation between them should be




(a) Directions used in approach $WS_1$. (b) Directions used in approach $WS_2$

Figure 7.4: $V_i, i = 1, 2, 3, 4$ represent motion vectors to be interpolated.

high. Thus, the weight calculation scheme above is unable to arrive at the correct conclusion. In this case, the weight is computed as $w = \max[(1 - \frac{||\mathbf{V_i}||}{th}), 0]$, where the threshold $th$ equals $\sqrt{50}$ which corresponds to one third of the general 15-pixel search range in each direction.

### 7.2.4 Simulation Results and Discussions

Simulations are performed to show the effectiveness of the proposed method in this section.

#### Simulation settings

**The encoder and GOP:** The baseline JPEG encoder is used to code residue error frames. A Group Of Picture (GOP) structure: IPPPP is used in the encoder, where each I frame is followed by four P frames. However, the improvements in compression performance is not limited by this GOP structure as typically, the larger the GOP size, the better the improvements. This is because the degradations in the reference frame propagate into the frames

which have used the degraded frame as a reference frame for motion estimation and motion compensation. Consequently, the quality differences of reconstructed frames (of the same frame index) in two competing methods become larger as the frame index goes towards the end of a GOP.

**Bit assignment scheme:** Bit assignment among I frames, motion vectors and residue error frames is implemented using a simple scheme. In this scheme, the quantization matrix associated with I frames is $\frac{1}{2}\mathbf{Q}$ while $\mathbf{Q}$ is the quantization matrix for the coding of residue error frames. The motion vectors are coded losslessly using the DPCM [109] scheme. Although more efficient but generally more complicated bit assignment schemes are available, it is not necessary to apply one in this simulation. This is because, in principle, a better bit assignment scheme would improve the performance of both repeat and the proposed methods.

**Parameters of the test video sequences:** The test sequences are encoded at 30fps, and 30 frames of each sequence are used in simulations. Sequences Foreman, Bus, Paris, Crew and Soccer are used. The comparison results can be interpreted as performance difference during one second at a 30 fps rate. The MPEG downsampling filter is used to obtain various spatial resolution sequences. The full spatial resolution is regarded as spatial layer 0. For 4CIF video sequences, CIF spatial resolution is obtained at spatial layer 1 and QCIF spatial resolution is obtained at spatial layer 2. For CIF video sequences, QCIF resolution is obtained at spatial layer 1 and SQCIF resolution is obtained at layer 2.

**Motion estimation parameters:** Full search motion estimation is performed at layer 2. The full search motion estimation parameters are: search range=4 pixels in each direction (which corresponds to 16 pixels in the highest resolution), motion vector precision= $\frac{1}{4}$pixel, and motion estimation block size=16×16 for QCIF spatial resolution, block size=8×8 for SQCIF spatial resolution.

**I frame coding:** Coding of I frames is not the focus of this simulation. However, if each of the three spatial layers is coded separately, too many bits would be used. To avoid this, the I-frame prediction coding scheme [104] is adopted here. In this scheme, only the base layer (layer 2) I frame is coded as a 'real' I frame while in the layer 1 and 0, I frames are predicted from layers 2 and 1, respectively, with the residue frames coded. The prediction frame is an interpolated version of the reconstructed I frame from the lower spatial layer.

### Coding of the mode maps

The experiment shows that one interpolation method would dominate the interpolation process for the test sequences. For example, around 60% blocks in the Foreman sequences are interpolated using $WS_1$ method, about 20% blocks are interpolated with the repeat method and the $WS_2$ and smoothing methods are used for the remaining blocks. Due to this observation, we fixed the mode maps coding scheme with 1 bit to indicate $WS_1$, 2 bits to indicate repeat method and 3 bits to represent $WS_2$ and smooth methods for all the test sequences.

We checked and compared the motion vector field and mode map coding at the 8th test point of the Foreman sequence. The average number of motion vector bits is around 130 bytes while the average number of mode map bits is around 80 bytes. The bits required to code the mode map is approximately $\frac{1}{6}$ of the amount that is required to code the corresponding motion vector field directly.

### Comparison of compression performances

The motion vectors obtained from full search motion estimation at layer 2 are interpolated using the repeat method and the proposed method to obtain the interpolated motion vector field for spatial layer 1 and 0, respectively. Motion compensation is performed according to the interpolated motion vector fields and residue frames are obtained by subtracting the motion compensated frames from the original frames. The motion vectors from spatial layer 2, the resulting error frames of three spatial layers and the mode maps (in the proposed method only)

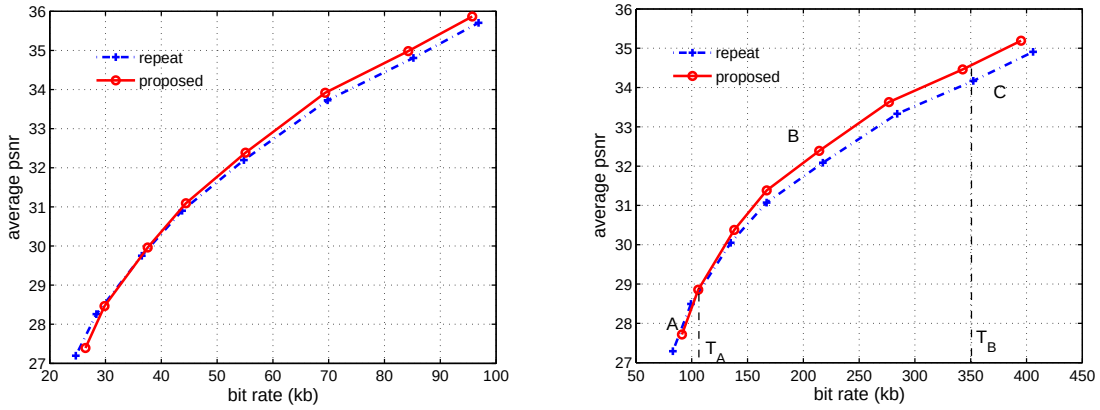
are coded. Then the total number of bits used in each method with respect to the PSNR quality of reconstructed frames is plotted.

The PSNR versus bitrate curves of the Foreman(CIF) and Soccer(4CIF) sequences are shown in Fig. 7.5 and Fig. 7.6 respectively. For discussion convenience, each curve is divided into three regions: region A, B and C. In region A the repeat method outperforms the proposed method. This is because at very low bitrate, very small amount of bits are assigned to code the residue error frames in both schemes. Although the proposed method consumes less bits, the bit difference is small. In this context, the actually small amount of bits that is used to code the mode maps in the proposed method becomes a significant percentage of the overall bitrate. In contrast, the PSNR improvement of the reconstructed frames is small. We checked the bit assignment results of the second test point, which is close to the separation bitrate $T_A$ in Fig. 7.5(b). Of the total 100 Kb in the repeat method, about 34 Kb are assigned to code the six I frames (total frames (30)/GOP size (5)=6) and about 3 Kb are used to code the 24 (No. of P frames) motion vector fields from the base layer. Thus, about 63 Kb are used to code the 24 three-layer residue error frames. On average, each three-layer error frame is assigned 2.6 Kb while the total data of it is about 133 Kb ($=352\times288\times(1+\frac{1}{4}+\frac{1}{16})$). Thus, the compression ratio of the error frame is about 48. This check verifies that a very small amount of bits is assigned to code the residue error frames in cases in region A (left to rate $T_A$). In regions B (between rate $T_A$ and $T_B$) and C (right to rate $T_B$ ), the proposed method outperforms the repeat method. We regard points in region C as cases where plenty of bits are available to code residue error frames. In these cases, if the encoder complexity isn't a big concern, performing full search motion estimation at each layer and coding motion vectors as well as low energy residue error frames from each layer is a better choice. Region B represents cases where the proposed method is a good choice: a low encoder complexity and a relatively low bitrate budget. The switching bitrate $T_B$ should be decided by considering complexity and bitrate budget jointly.

For the same reconstruction quality, the typical bits saved are 10% as highlighted in Fig. 7.6(b) where the bitrate difference, $S$, is more than 100 Kb, which

is more than 10% of the total 1000 Kb.

In addition to the Foreman and Soccer sequences, we test video sequences Bus(CIF), Paris(CIF) and Crew(4CIF) and observe that the proposed method performs consistently better than the repeat method for various video sequences.




(a) at layer 1

(b) at layer 0

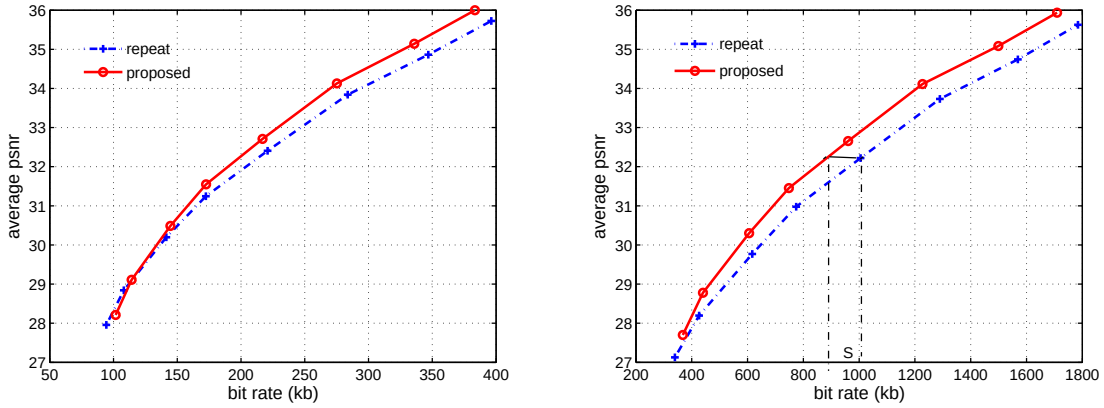
Figure 7.5: Reconstruction quality versus bitrate (Foreman). Layer 2 is the base layer and the spatial resolution is SQCIF.

### 7.2.5 Conclusions

We presented a class of weighted smoothing methods for inter-layer motion vector interpolation and a mode map method to code the interpolated motion vector fields efficiently. The two techniques together provide a low complexity and low bitrate scalable video coding scheme.

The weights in the weighted smoothing methods measure the correlations of neighboring motion vectors and the weighted smoothness is maximized thus avoiding smoothness at motion boundaries. This class of methods model the naturally smooth but local motions in real video sequences.

The mode map for each spatial layer is generated to indicate the chosen interpolation method for a particular block and can be coded very efficiently. Future work will be on reducing the overhead of the mode map by exploiting inter-layer correlation.




(a) at layer 1

(b) at layer 0

Figure 7.6: Reconstruction quality versus bitrate (Soccer). Layer 2 is the base layer and the spatial resolution is SQCIF.

The text of this chapter is adapted from the material that has published in part as: M. Li and T. Q Nguyen, *Optimal wavelet filter design in scalable video coding*, *IEEE International Conference on Image Processing* 2005 and also will appear in part as M. Li, P. Chandrasekhar, G. Dane and T. Q Nguyen,*Low-Complexity and Low Bitrate Scalable Video Coding Scheme Using Inter-Layer Motion Vector Interpolation Techniques*, *Asilomar Conference on Signals, Systems and Computers* 2007. The dissertation author was the primary researcher of these publications, and the co-author T. Q Nguyen listed directed and supervised the research which forms the basis for this chapter.

# 8

# Conclusions and Future Work

With the development and improvements of digitization and display technologies, digital video broadcasting evolves into the broadcasting of multimedia content to both homes and mobile devices. During this evolution, the requirements on image qualities have been increased significantly, which call for efficient post-processing algorithms.

Furthermore, with the ever increasing computational power of today's processors, it becomes feasible to use more robust and computationally more complex algorithms to perform content-adaptive post-processing. The traditional algorithms have been preferred not for their performance but for their low implementation complexity. The work in this dissertation explores the application of Markov Random Field (MRF) models in video post processing. MRF models are able to model the spatial dependencies among pixels in an image and can serve as a tool to impose spatial constraints on the processed images. It is highly locally adaptive and is a state-of-art fit for the content-adaptive processing purpose.

The work done in this thesis is summarized as follows.

**Formulation of 2-D DAS Constraint**

We formulate the 2-D discontinuity-adaptive smoothness constraint and impose it on the processed images via a MRF model. This constraint assumes piecewise smoothness of images, where the smoothness does not cross discontinuity features,

such as edges and object boundaries. This helps the processed image to achieve smoothness along edge directions and remain sharp in across-edge directions.

**Statistic-based approach to measure local edge direction information**

Local edge direction information is required when formulating the 2-D DAS constraint. Considering the limitations of the conventional binary representation (edge direction, non-edge direction) of local edge directions in natural images, we proposed a statistic-based approach to measure the local edge direction. That is more robust. In this approach, the local edge direction is represented by a weight vector. Each entry of the vector corresponds to a weight that is associated with a discrete direction and a relatively large weight indicates a higher possibility of being the local edge direction.

**Application in MC de-interlacing**

The formulated 2-D DAS constraint is applied to MC de-interlacing and spatial interpolation problems. In MC de-interlacing, efficient protection against erroneous MVs is necessary. Furthermore, the protection should not be over-protective, which could limit the advantages of MC. Thus, it is challenging for existing protection strategies to simultaneously guarantee consistency and sharpness of edges. Using a MRF model, the 2-D DAS constraint is imposed on the de-interlaced frame. The final output of each pixel site will be the one that fits its context best under the 2-D DAS constraint. The context includes neighboring pixels from both the available field and newly interpolated field. The de-interlaced edges are expected to have strong geometric regularity.

**MRF-EDI spatial interpolation**

As for spatial interpolation methods, the traditional polynomial-based interpolation methods assume continuity of the pixel intensity field, which is unrealistic, and typically produce images with blurred edges. Edge-directed interpolation is proposed to improve the interpolation performance. However, it is challenging to specify the edge direction explicitly because edges in natural images are widely

contaminated by all kinds of imaging noise. We proposed the MRF-EDI interpolation method, which uses the local edge direction information through the 2-D DAS constraint in an implicit manner. The interpolated image is clean and has sharper edges.

**Low complexity implementation of the proposed algorithms**

The proposed MRF-based MC de-interlacing and spatial interpolation algorithms are iterative optimization algorithms. The computational complexity may be too high for real-time processing. We investigated the low-complexity implementation of the algorithms. One strategy is to limit the proposed processing only to pixels near edge areas where a pre- discrimination process discriminates edge pixels from smooth area pixels. We also designed a *single pass* implementation to remove the iterations in the algorithm. The degradation in the interpolation results has been minimal.

In addition to the work with video post-processing, this thesis also contains some work with wavelet-based and H.264 AVC- based scalable video coding.

**Filters design in wavelet-based SVC**

In wavelet-based SVC, the Low Band Correction (LBC) technology is preferred for its efficiency in reducing overhead information and the size of the bit stream. However, with the LBC techonology, the lowpass filter implementing spatial scalability has to be a wavelet filter, which obeys the strong half-band condition. The requirements of being a good wavelet filter for compression purposes and of being a good lowpass filter for scalability purpose are conflicting requirements.

Using the maximally flat wavelet filters as prototype filters, we develop a parametric design method to design a class of wavelet filters. The design method incorporates all of the wavelet filter design requirements and achieves optimal tradeoffs of conflicting requirements. Consequently, the designed filters are suitable for both motion estimation and compression in SVC.

**Motion vector prediction in low-complexity low bit rate H.264-based SVC**

In the H.264 AVC-based scalable video model, motion vectors have to be coded and sent for all the spatial layers even under low complexity and low bit-rate requirements, which is not efficient.

We investigated the possibility of performing inter-layer motion vector prediction/interpolation in low bit-rate applications such that not all motion vectors have to be sent to the decoder. The performance of several methods was tested and compared. There is no single method that performs constantly better than others and thus we combined the interpolation methods to produce a novel mode-map to indicate the chosen method for a block. The mode map can be coded efficiently and the overall inter-layer motion vector prediction/interpolation method does not smooth motion vectors that are near motion boundaries.

In the future, this work can be extended in the following two aspects.

1) The proposed MRF-based algorithm can be modified to perform other video post-processing, such as edge-enhancement (sharpening) and motion de-blurring. For example, in the sharpening application, the proposed MRF model can function as a regularizer to optimize the sharpened images.

2) The proposed algorithms can be modified to have better content-adaptive properties. Currently, all local discontinuity features are dealt with as straight lines, which is not accurate enough for corners and curves with large curvatures. If a better model or approximation of corners and arbitrary curves can be formulated, the algorithm should demonstrate an even stronger content-adaptive property. We hope this work can inspire more efficient MRF-based video post processing algorithms, which can show stronger content-adaptive property.

3) After the defensing presentation, one committee member pointed out that the robustness of the potential function is one important aspect to study. I

agree with this point and include the robustness analysis of potential functions as one of my future works. One reference paper for this kind of analysis is [110], which analyzed the outlier rejection property of line processes.

# 9

# Appendices

## 9.1 Appendix A: The Half-Normal Distribution

Let the random variable $Z$ have a zero mean normal distribution, $Z \sim N(0,1)$, and let $W$ equal the absolute value of $Z$, $W = |Z|$. Then the random variable $W$ obeys the half normal distribution [111] and the probability density function of $W$ can be expressed as

$$f(w) = \frac{2}{\sqrt{2\pi}} \exp\{-\frac{w^2}{2}\}, w \geq 0. \tag{9.1}$$

The expectation of $W^2$, E[$W^2$], equals E[$|Z|^2$]=E[$Z^2$] = 1 and the expectation of $W$ can be obtained by integrating $wf(w)$ from 0 to $\infty$. The result is constant $\lambda = \sqrt{\frac{2}{\pi}}$.

If random variable $X$ obeys the general normal distribution, $X \sim N(\mu, \sigma^2)$, random variable $Y = \mu + |X - \mu|$ obeys the general half normal distribution and the probability density function can be expressed as

$$f(y) = \frac{2}{\sqrt{2\pi}\sigma} \exp\{-\frac{(y-\mu)^2}{2\sigma^2}\}, y \geq \mu. \tag{9.2}$$

The expectation values of $Y$ and $Y^2$ are E[$Y$] = $\lambda\sigma + \mu$ and E[$Y^2$] = $\sigma^2 + \mu^2 + 2\lambda\sigma\mu$ respectively, which can be derived by employing the facts $Z = \frac{X-\mu}{\sigma}$, $W = \frac{Y-\mu}{\sigma}$ and linear properties of the expectation operation.

By setting the mean $\mu$ as 0, it can be concluded that, if random variable $X$ obeys distribution $N(0, \sigma^2)$, variable $Y = |X|$ will obey half normal distribution with E[$Y$] = $\lambda\sigma$ and E[$Y^2$] = $\sigma^2$.

## 9.2 Appendix B: The Gibbs Distribution and the MRF Property

Suppose $P(\omega)$ is a Gibbs distribution on state space $\mathcal{S}$ with respect to neighborhood structure $\mathcal{N}$. The conditional probability of pixel $i$ given all the other pixels in the configuration, which is represented by $\omega_{\mathbf{I}-\{i\}}$ , is

$$P(\omega_i|\omega_{\mathbf{I}-\{i\}}) = \frac{P(\omega_i, \omega_{\mathbf{I}-\{i\}})}{P(\omega_{\mathbf{I}-\{i\}})} = \frac{P(\omega)}{\sum_{\omega_i' \in \omega'} P(\omega_i')}, \tag{9.3}$$

where $\omega' = \{\omega_1', \cdots, \omega_{i-1}', \omega_i', \cdots, \omega_M'\}$ is any configuration which agrees with $\omega$ at all sites except for pixel site $i$.

As $P(\omega)$ obeys Gibbs distribution, we have

$$P(\omega) = \frac{\exp\{-\sum_{c\in\mathcal{C}} V_c(\omega)\}}{Z}, \tag{9.4}$$

From (9.3) and (9.4), we obtain

$$P(\omega_i|\omega_{\mathbf{I}-\{i\}}) = \frac{\exp\{-\sum_{c\in\mathcal{C}} V_c(\omega)\}}{\sum_{\omega_i'\in\omega'} \exp\{-\sum_{c\in\mathcal{C}} V_c(\omega_i')\}}. \tag{9.5}$$

The cliques set $\mathcal{C}$ can be written as the union of two subsets, $\mathcal{C} = \mathcal{A} \cup \mathcal{B}$, where $\mathcal{A}$ includes only those cliques that contain site $i$ and $\mathcal{B}$ includes all the other cliques that do not contain site $i$. With $\mathcal{C} = \mathcal{A} \cup \mathcal{B}$, the conditional probability can be expressed as

$$P(\omega_i|\omega_{\mathbf{I}-\{i\}}) = \frac{\exp\{-\sum_{c\in\mathcal{A}} V_c(\omega)\} \exp\{-\sum_{c\in\mathcal{B}} V_c(\omega)\}}{\sum_{\omega_i'\in\omega'} \exp\{-\sum_{c\in\mathcal{A}} V_c(\omega_i')\} \exp\{-\sum_{c\in\mathcal{B}} V_c(\omega_i')\}}. \tag{9.6}$$

Because $V_c(\omega) = V_c(\omega')$ for any clique $c$ that does not contain $i$, $\exp\{-\sum_{c\in\mathcal{B}} V_c(\omega)\}$ is cancelled out by $\exp\{-\sum_{c\in\mathcal{B}} V_c(\omega_i')\}$. Eventually, the conditional probability is expressed as

$$P(\omega_i|\omega_{\mathbf{I}-\{i\}}) = \frac{\exp\{-\sum_{c\in\mathcal{A}} V_c(\omega)\}}{\sum_{\omega_i'\in\omega'} \exp\{-\sum_{c\in\mathcal{A}} V_c(\omega_i')\}}. \tag{9.7}$$

That is, the conditional probability depends on $i$'s neighbors only. Thus, a GD results in an MRF.

# Bibliography


[1] Y. Zhang and K.-K. Ma, "Error concealment for video transmission with dual multiscale markov random field modeling," *IEEE Trans. on Image Processing*, vol. 12, no. 2, pp. 236–242, Feb. 2003.

[2] S. Geman and D. Geman, "Stochastic relaxation, gibbs distribution, and the Bayesian restoration of images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721–741, Nov. 1984.

[3] S. Z. Li, "On discontinuity-adaptive smoothness priors in computer vision," *IEEE Trans. on Pattern and Machine Intelligence*, vol. 17, no. 6, pp. 576–586, June 1995.

[4] U. Horn, R. Keller, and N. Niebert, "Interactive mobile streaming services-the convergence of broadcast and mobile communication," *European Broadcasting Union Technical Review*, 1999.

[5] U. Reimers, *DVB: The Family of International Standards for Digital Video Broadcasting*, 2nd ed., ser. Signals and Communication Technology. Berlin Heidelberg: Springer, 2005.

[6] C. Weck and E. Wilson, "Broadcasting to handhelds-an overview of systems and services," *European Broadcasting Union Technical Review*, Jan. 2006.

[7] O. Werner, "Broadcasters' requirements for IPTV," *European Broadcasting Union Technical Review*, Apr. 2007.

[8] E. M. Schwalb, *iTV Handbook Technologies and Standards*, ser. IMSC Press Multimedia. Upper Saddle River, NJ: Prentice Hall, 2004.

[9] J. Whitaker, *Mastering Digital Television: the Complete Guide to the DTV Conversion*, ser. Video/Audio Professional. United States of America: McGraw-Hill's, 2006.

[10] M. Biswas, "Content adaptive video processing algorithms for digital TV," PhD Dissertation, UC San Diego, Electrical and Computer Engineering Department, Aug. 2005.

[11] R. A. Beuker and I. A. Shah, "Analysis of interlaced video signals and its applications," *IEEE Trans. on Image Processing*, vol. 3, no. 5, pp. 501–512, Sept. 1994.

[12] S. Hartwig, "On the performance bounds of motion-compensated deinterlacing," *Signal Processing: Image Communication*, vol. 10, no. 4, pp. 291–302, Sept. 1997.

[13] K. J. Kuo, C. Liao, and C. C. Lin, "Adaptive interpolation technique for scanning rate conversion," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 6, no. 3, June 1996.

[14] S. F. Lin, Y. L. Chang, and L. G. Chen, "Motion adaptive interpolation with horizontal motion detection for de-interlacing," *IEEE Trans. on Consumer Electronics*, vol. 49, no. 4, pp. 1256–1265, Nov. 2003.

[15] E. Dubois, G. de Haan, and T. Kurita, "Motion estimation and compensation technologies for standards conversion," *Signal processing. Image communication*, 1994.

[16] D. Hargreaves and J. Vaisey, "Bayesian motion estimation and interpolation in interlaced video sequences," *IEEE Trans. on Image Processing*, vol. 6, no. 5, pp. 764–769, May 1997.

[17] M. Biswas and T. Nguyen, "A novel de-interlacing technique based on phase plane correlation motion estimation," *Proceedings of IEEE International Symputm on Circuits and Systems*, vol. 2, 2003.

[18] E. B. Bellers and G. de Haan, *De-Interlacing: a Key Technology for Scan Rate Conversion*. Netherlands: Elsevier Science B.V., 2000.

[19] H. Yoo and J. Jeong, "Direction-oriented interpolation and its application to de-interlacing," *IEEE Trans. on Consumer Electronics*, vol. 48, no. 4, pp. 954–962, Nov. 2002.

[20] P. Brox, I. Baturone, and S. Sanchez-Solano, "Interlaced to progressive scan conversion using fuzzy edge-based line average algorithm," *IEEE International Workshop on Intelligent Signal Processing*, pp. 10–15, Sept. 2005.

[21] N. Luo, Y. Chai, and X. Fang, "A median filtering de-interlacing algorithm convenient for hardware implementation algorithm," *IEEE International Symposium on Communications and Information Technology*, vol. 1, pp. 453–456, Oct. 2005.

[22] G. de Haan and E. B. Bellers, "De-interlacing-an overview," *Proceedings of IEEE*, vol. 86, no. 9, pp. 1839–1857, Sept. 1998.

[23] S.-K. Kwon, K.-S. Seo, J.-K. Kim, and Y.-G. Kim, "A motion-adaptive de-interlacing method," *IEEE Trans. on Cons. Electr.*, vol. 38, no. 3, pp. 145–150, Aug. 1992.

[24] S.-G. Lee and D.-H. Lee, "A motion-adaptive de-interlacing method using an efficient spatial and temporal interpolation," *IEEE Trans. on Cons. Electr.*, vol. 49, no. 4, pp. 1266–1271, Nov. 2003.

[25] Genesis, "gmVLD8, 8 bit digital video line doubler," *Proceedings of IEEE*, Jan. 1997, version 2.

[26] Philips, "SAA4990H, progressive scan, zoom, and noise reduction IC," *Proceedings of IEEE*, Oct. 1996.

[27] G. de Haan, "IC for motion-compensated de-interlacing, noise reduction, and picture-rate conversion," *IEEE Trans. on Consumer Electronics*, vol. 45, no. 3, pp. 617–624, Aug. 1999.

[28] G. de Haan, P. Biezen, H. Huijgen, and O. A. Ojo, "True motion estimation with 3-D recursive search block-matching," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 3, no. 5, pp. 368–388, Oct. 1993.

[29] G. de Haan and P. Biezen, "Sub-pixel motion estimation with 3-D recursive search block-matching," *signal Processing: Image Communication*, vol. 6, no. 3, pp. 229–239, June 1994.

[30] M. Biswas, S. Kumar, and T. Nguyen, "Performance analysis of motion compensated de-interlacing systems," *IEEE Trans. on Image Processing*, vol. 15, no. 9, pp. 2596–2609, Sept. 2006.

[31] Y.-L. Chang, S.-F. Lin, C.-Y. Chen, and L.-G. Chen, "Video de-interlacing by adaptive 4-field global/local motion compensated approach," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 15, no. 12, pp. 1569–1582, Dec. 2005.

[32] L. Yin, R. Yang, M. Gabbouj, and Y. Neuvo, "Weighted median filters: a tutorial," *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 43, no. 3, pp. 157–192, Mar. 1996.

[33] P. Haavisto, J. Juhola, and Y. Neuvo, "Fractional frame rate up-conversion using weighted median filters," *IEEE Trans. on Cons. Electr.*, vol. 35, no. 3, pp. 272–278, Aug. 1989.

[34] G. de Haan, *Video Processing for Multimedia Systems.* Eindhoven: University Press, 2000, ch. 5.

[35] H. Hou and H. Andrews, "Cubic splines for image interpolation and digital filtering," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 26, no. 6, pp. 508–517, Dec. 1978.

[36] M. Sakalli, Y. Hong, and A. M. N. Fu, "A fuzzy-Bayesian approach to image expansion," *International Joint Conference on Neural Networks*, vol. 4, pp. 2685–2689, July 1999.

[37] W. K. Carey, D. B. Chuang, and S. S. Hemami, "Regularity-preserving image interpolation," *IEEE Trans. on Image Processing*, vol. 8, no. 9, pp. 1293–1297, Sept. 1999.

[38] 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.

[39] 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.

[40] Q. Wang and R. Ward, "A new edge-directed image expansion scheme," *Proceedings of International Conference on Image Processing*, vol. 3, pp. 899–902, Oct. 2001.

[41] X.-Q. Chen, J. Zhang, and L.-N. Wu, "Improvement of a nonlinear image interpolation method based on heat diffusion equation," *International Conference on Machine Learning and Cybernetics*, vol. 5, pp. 2911–2914, Nov. 2003.

[42] S. Lee and J. Paik, "Image interpolation using adaptive fast B-spline filtering," *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 5, pp. 177–180, Apr. 1993.

[43] A. Gotchev, K. Egiazarian, J. Vesma, and T. Saramaki, "Edge-preserving image resizing using modified B-splines," *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 3, pp. 1865–1868, May 2001.

[44] K. Jensen and D. Anastassiou, "Subpixel edge localization and the interpolation of still images," *IEEE Trans. on Image Processing*, vol. 4, pp. 285–295, Mar. 1995.

[45] J. Allebach and P. Wong, "Edge-directed interpolation," *IEEE International Conference on Image Processing*, vol. 3, pp. 707–710, Sept. 1996.

[46] J. Canny, "A computational approach to edge detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679–698, 1986.

[47] F. A. Pellegrino, W. Vanzella, and V. Torre, "Edge detection revisited," *IEEE Trans. on Systems, Man and Cybernetics-Part B: Cybernetics*, vol. 34, no. 3, pp. 1500–1511, June 2004.

[48] J. H. Elder and S. W. Zucker, "Local scale control for edge detection and blur estimation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 7, pp. 699–716, July 1998.

[49] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Trans. on Image Processing*, vol. 10, no. 10, pp. 1521–1526, 2001.

[50] J. Besag, "Nearest-neighbor systems and the auto-logistic model for binary data," *J. Royal Statist. Soc.*, vol. 34, pp. 75–83, 1972.

[51] I. Manno, *Introduction to the Monte Carlo Method*. Hungary: Akademiai Kiado Budapest, 1999.

[52] H. Rue and L. Held, *Gaussian Markov Random Fields : Theory and Applications*. Boca Raton: Chapman & Hall/CRC, Taylor & Francis Group, 2005.

[53] S. G. Nadabar and A. K. Jain, "Parameter estimation in MRF line process models," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 528–533, June 1992.

[54] E. T. Jaynes, "Information theory and statistical mechanics," *Phys. Rev.*, vol. 106, no. 4, pp. 620–630, May 1957.

[55] H. Bass, J. Kingman, F. S. J. todd, and C. Wall, *Gibbs states on countable sets*. UK: Cambridge University Press, 1974.

[56] C. J. Preston, "Generalized Gibbs states and Markov Random Fields," *Advances in Applied probability*, vol. 5, 1973.

[57] R. Kinderman and J. Snell, *Marokv Random Fields and Their Applications*. Providence, RI: Amer. Math. Soc., 1980.

[58] J. Besag, "Spatial interaction and the statistical analysis of lattice systems (with discussion)," *J. Royal Statist. Soc.*, vol. 36, pp. 192–326, 1974.

[59] B. Girod, "The efficiency of motion-compensating prediction for hybrid coding of video sequences," *IEEE Journal on Selected Areas in Communications*, no. 7, pp. 1140–1154, Aug. 1987.

[60] P.-L. Shui, "Image denoising algorithm via doubly local Wiener filtering with directional windows in wavelet domain," *IEEE Signal Processing Letters*, vol. 12, no. 10, pp. 681–684, Oct. 2005.

[61] M. Kazubek, "Wavelet domain image denoising by thresholding and Wiener filtering," *IEEE Signal Processing Letters*, vol. 11, no. 10, pp. 324–326, Nov. 2003.

[62] F. Jin, P. Fieguth, L. Winger, and E. Jernigan, "Adaptive Wiener filtering of noisy images and image sequences," *Proceedings of International Conference on Image Processing*, vol. 3, pp. III.349–III.352, Sept. 2003.

[63] R. Prendergast and T. Nguyen, "A non-isotropic parametric model for image spectra," *IEEE Proceedings of International Conference on Acoustics, Speech and Signal Processing*, vol. 2, pp. II.761–II.764, May 2006.

[64] ——, "Improving frequency doamin super-resolution via undersampling model," *IEEE Proceedings of International Conference on Image Processing*, vol. 1, pp. 853–856, Sept. 2005.

[65] Y. Xu, J. B. Weaver, D. M. Healy, and J. Lu, "Wavelet transform domain filters: a spatially selective noise filtration technique," *IEEE Trans. on Image Proc.*, vol. 3, 1994.

[66] G. P. Nason, "Choice of wavelet smoothness, primary resolution and threshold in wavelet shrinkage," *Statistics and Computing*, vol. 12, no. 3, pp. 219–227, July 2002.

[67] S. Mallat and W. L. Hwang, "Singularity detection and processing with wavelets," *IEEE Trans. Information Theory*, vol. 38, pp. 617–643, 1992.

[68] H. Xie, L. E. Pierce, and F. T. Ulaby, "SAR speckle reduction using wavelet denoising and markov random field modeling," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 40, no. 10, pp. 2196–2211, Oct. 2002.

[69] M. Malfait and D. Roose, "Wavelet-based image denoising using a markov random field a priori model," *IEEE Trans. on Image Proc.*, vol. 6, no. 4, pp. 549–565, Apr. 1997.

[70] M. Jansen and A. Bultheel, "Empirical Bayes approach to improve wavelet thresholding for image noise reduction," *Journal of the American Statistical Association*, vol. 96, no. 454, pp. 629–639, June 2001.

[71] A. Pizurica, W. Philips, I. Lemahieu, and M. Acheroy, "A joint inter- and intrascale statistical model for Bayesian wavelet based image denoising," *IEEE Trans. on Image Proc.*, vol. 11, 2002.

[72] P. Bouthemy, C. Hardouin, G. Piriou, and J. Yao, "Mixed-state auto-models and motion texture modeling," *Journal of Mathematical Imaging*, vol. 25, no. 3, pp. 387–402, 2006.

[73] S. C. Zhu, "Filters, random fields and maximum entropy (FRAME): towards a unified theory for texture modeling," *International Journal of Computer Vision*, vol. 27, no. 2, pp. 107–126, 1998.

[74] S. C. Zhu, Y. N. Wu, and D. Mumford, "minimax entropy principle and its application to texture modeling," *Neural Computations*, vol. 9, 1997.

[75] S. Z. Li, H. Wang, and M. Petrou, "Relaxation labeling of Markov Random Fields," *Proceedings of the 12th IAPR International Conference on Pattern Recognition, Conference A: Computer Vision & Image Processing*, vol. 1, pp. 488–492, Oct. 1994.

[76] Y. Xia, D. G. Feng, and R. C. Zhao, "Adaptive segmentation of textured images by using the coupled markov random field model," *IEEE Trans. on Image Proc.*, vol. 15, no. 11, pp. 3559–3566, Nov. 2006.

[77] T. Crivelli, B. Cernuschi-Frias, P. Bouthemy, and J. Yao, "Mixed-state markov random fields for motion texture modeling and segmentation," *International Conference on Image Processing*, 2006.

[78] Z. Li and E. J. Delp, "Block artifact reduction using a transform-domain markov random field model," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 15, no. 12, pp. 1583–1593, Dec. 2005.

[79] M. N. Chong and D. Krishnan, "An edge-preserving MRF model for the detection of missing data in image sequences," *Signal Processing Letters*, vol. 5, no. 4, pp. 81–83, Apr. 1998.

[80] P. J. Huber, *Robust Statistics*. New York: Wiley, 1981.

[81] S. Z. Li, *Markov Random Field Modeling in Image Analysis*. Springer-Verlag, 2001.

[82] S. G. Nadabar and A. K. Jain, "Parameter estimation in MRF linear process models," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 14, no. 6, pp. 528–533, June 1992.

[83] S. Z. Li, "Reconstruction without discontinuities," *International Conference on Computer Vision*, pp. 709–712, Dec. 1990.

[84] E. W. Weisstein, "Euler-lagrange differential equation," *From MathWorld–A Wolfram Web Resource*, http://mathworld.wolfram.com/Euler-LagrangeDifferentialEquation.html.

[85] M. K. Park, M. G. Kang, K. Nam, and S. G. Oh, "New edge dependent de-interlacing algorithm based on horizontal edge pattern," *IEEE Trans. on Consumer Electronics*, vol. 49, no. 4, pp. 1508–1512, Nov. 2003.

[86] N. Madras, *Lectures on Monte Carlo Methods.* American Mathematical Society, 2002, pp. 69–81.

[87] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *J. Chem. Phys.*, vol. 21, pp. 1087–1092, 1953.

[88] M. Li and T. Nguyen, "Discontinuity-adaptive de-interlacing scheme using markov random field model," *IEEE International Conference on Image Processing*, pp. 393–396, Oct. 2006.

[89] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Transaction on Computer Graphics and Application*, vol. 22, no. 2, pp. 56–65, Apr. 2002.

[90] X. Li, http://www.csee.wvu.edu/∼xinl/source.html.

[91] J. R. Ohm, "Three-dimensional subband coding with motion compensation," *IEEE Trans. on Image Proc.*, vol. 3, no. 5, Sept. 1994.

[92] S. J. Choi and J. W. Woods, "Motion compensated 3-D subband coding of video," *IEEE Trans. on Image Proc.*, vol. 8, no. 2, pp. 155–167, Feb. 1999.

[93] ISO/IEC JTC1, "Requirements and applications for scalable video coding," *ISO/IEC JTC1/WG11/N6025*, Oct. 2003.

[94] M. Asia IM group and M. C. group, "3D sub-band video coding using barbell lifting," *ISO/IEC JTC1/SC29/WG11 MPEG2004/M10569/S05*, Mar. 2004.

[95] W.-J. Han, "Responses of call-for-proposal for scalable video coding," *ISO/IEC JTC 1/SC 29/WG11 MPEG2004/M10569/S17*, 2004.

[96] H. Schwarz, D. Marpe, and T. Wiegand, "Scalable extension of H.264/AVC," *ISO/IEC JTC 1/SC 29/WG11 MPEG2004/M10569/S03*, 2004.

[97] MPEG Adhoc Group, "AhG on further exploration in wavelet video coding," *ISO/IEC JTC 1/SC 29/WG11 MPEG2004/N6732*, Oct. 2004.

[98] W. Li, J.-R. Ohm, M. van der Schaar, H. Jiang, and S. Li, "MPEG-4 video verification model version 18.0," *ISO/IEC JTC 1/SC 29/WG11 MPEG2001/N3908*, 2001.

[99] M. Unser and T. Blu, "Mathematical properties of the JPEG-2000 wavelet filters," *IEEE Trans. on Image Proc.*, vol. 12, no. 9, pp. 1080–1090, Sept. 2003.

[100] G. Strang and T. Nguyen, *Wavelet and Filter Banks.* P T R, New Jersey: Wellesley-Cambridge Press, 1996.

[101] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *J. Fourier Anal. Appl.*, vol. 4, no. 3, pp. 247–269, 1998.

[102] I. Daubechies, *Ten Lectures on wavelets.* Philadelphia: PA: SIAM, 1992, ch. 9.

[103] P. Vaidyanathan and T. Nguyen, "Eigenfilters: A new approach to least-squares FIR filter design and applications including nyquist filters," *IEEE Transactions on Circuits and Systems*, vol. 34, no. 1, pp. 11–23, Jan. 1987.

[104] ISO/IEC JTC1, "Scalable Video Model 3.0 draft," *ISO/IEC JTC1/SC 29/WG11 N6716*, 2004.

[105] Y. Nie and K.-K. Ma, "Adaptive root pattern search for fast block-matching motion estimation," *IEEE Trans. on Image Proc.*, vol. 11, no. 12, pp. 1442–1449, Dec. 2002.

[106] G. Dane and T. Nguyen, "Smooth motion vector resampling for standard compatible video post-processing," *Asil. Conf. on Sign., Syst. & Comp.*, Nov. 2004.

[107] M.-G. Liu and C.-H. Hou, "A fast block-matching motion estimation algorithm based on spatial-temporal motion vector correlation," *International Symposium on Intelligent Multimedia, Video and Speech Processing*, pp. 498–501, May 2005.

[108] C. H. Hsieh, P. Lu, J. S. Shyn, and E.-H. Lu, "Motion estimation algorithm using interblock correlation," *Electronics Letters*, pp. 276–277, Mar. 1990.

[109] J. Watkinson, *MPEG-2.* Boston: Focal Press, 1999.

[110] M. Black, "On the unification of line processes, outlier rejection, and robust statistics with applications in early vision," *International Journal of Computer Vision*, vol. 19, no. 1, pp. 57–91, 1996.

[111] R. Hogg and E. Tanis, *Probability and Statistical Inference*, 6th ed. Upper Saddle River, NJ: Prentice Hall, 2001.