UNIVERSITY OF CALIFORNIA, SAN DIEGO

# Motion Vector Processing in Compressed Video and Its Applications to Motion Compensated Frame Interpolation

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

in

Electrical and Computer Engineering

by

Ai-Mei Huang

Committee in charge:

Professor Truong Nguyen, Chair
Professor Pamela C. Cosman
Professor Rajesh Gupta
Professor William S. Hodgkiss
Professor Nuno Vasconcelos

2009

The dissertation of Ai-Mei Huang is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2009

## DEDICATION

To my husband, my daughter, and my son

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# ACKNOWLEDGEMENTS

First of all, I would like to express gratitude to my advisor Dr. Truong Nguyen. Dr. Nguyen is always ready to discuss my work, and also let me free to pursue my own goals in my own way. Without his guidance, I would not be able to successfully complete this dissertation. It has been a great pleasure to have him as a supervisor. I also thank Dr. Cosman, Dr. Hodgkiss, Dr. Vasconcelos, and Dr. Gupta for their input and guidance serving in my Ph.D. committee.

More in general, my years at the Video Processing Lab will remain a pleasant memory for its excellent working environment and friendly atmosphere. Finally, I would like to express the deepest gratitude to my parents and my husband for giving me so much in everything so that I can concentrate on doing research and successfully complete this thesis.

Portions of Chapter 2 appear in “A Multistage Motion Vector Processing Method for Motion-Compensated Frame Interpolation,” in *IEEE Transactions on Image Processing*, May 2008 and “Correlation-Based Motion Vector Processing with Adaptive Interpolation Scheme for Motion Compensated Frame Interpolation,” in *IEEE Transactions on Image Processing*, April 2009. The dissertation author was the primary author of these publications, and the listed co-author directed and supervised the research that forms the basis for this chapter.

Portions of Chapter 3 appear in “A Multistage Motion Vector Processing Method for Motion-Compensated Frame Interpolation,” in *IEEE Transactions on Image Processing*, May 2008, “Motion Vector Processing Based on Residual Energy Information for Motion Compensated Frame Interpolation,” in *IEEE International Conference on Image Processing*, Oct. 2006, “A Novel Motion Compensated Frame Interpolation Based on Block-Merging and Residual Energy” in *IEEE Workshop on Multimedia Signal Processing*, Oct. 2006, “A Novel Multi-Stage Motion Vector Processing Method for Motion Compensated Frame Interpolation,” in *IEEE International Conference on Image Processing*, May 2007, and “Motion Vector Processing Using Bidirectional Frame Difference in Motion Compensated Frame Interpolation,” in *IEEE Workshop on Mobile Video Delivery*, June 2008. The dissertation author was the primary author of these publications, and the listed co-author directed and supervised the research that forms the basis for this chapter.

Portions of Chapter 4 appear in “Correlation-Based Motion Vector Processing with Adaptive Interpolation Scheme for Motion Compensated Frame Interpolation,” in *IEEE Transactions on Image Processing*, April 2009, and “Correlation-based Motion Vector Processing For Motion Compensated Frame Interpolation,” in *IEEE International Conference on Image Processing*, Oct. 2008. The dissertation author was the primary author of these publications, and the listed co-author directed and supervised the research that forms the basis for this chapter.

Portions of Chapter 5 appear in “Correlation-Based Motion Vector Processing with Adaptive Interpolation Scheme for Motion Compensated Frame Interpolation,” in *IEEE Transactions on Image Processing*, April 2009, and “Correlation-based Motion Vector Processing For Motion Compensated Frame Interpolation,” in *IEEE International Conference on Image Processing*, Oct. 2008. The dissertation author was the primary author of these publications, and the listed co-author directed and supervised the research that forms the basis for this chapter.

I am very grateful for the financial support of Conexant Systems, Incorporated, Nvidia Corporation, and the University of California Discovery Grant. I hope I have represented their contributions well.

## VITA

| | |
|---|---|
| 1999 | B. S. in Communication Engineering, National Chiao Tung University, Taiwan |
| 2000 | M. S. in Electrical Engineering, National Taiwan University, Taiwan |
| 2009 | Ph. D. in Electrical and Computer Engineering, University of California, San Diego |

## PUBLICATIONS

### Journal Publications:

1. A.-M. Huang and T. Nguyen, "True Motion Processing Based on Motion Trajectory Curve Analysis and Its Application to Motion Compensated Frame Interpolation," submitted to *IEEE Transactions on Image Processing*, April 2009.
2. A.-M. Huang and T. Nguyen, "Correlation-Based Motion Vector Processing with Adaptive Interpolation Scheme for Motion Compensated Frame Interpolation," in *IEEE Transactions on Image Processing, Vol. 18, issue 4, pp. 740-752*, April 2009.
3. K.-C. Yang, A.-M. Huang, T. Nguyen, C.C. Guest and P.K. Das, "A New Objective Quality Metric for Frame Interpolation Used in Video Compression," in *IEEE Transactions on Broadcasting, vol. 54, issue 3, Part 2, pp. 680-690*, September 2008.
4. A.-M. Huang and T. Nguyen, "A Multistage Motion Vector Processing Method for Motion-Compensated Frame Interpolation," in *IEEE Transactions on Image Processing, vol. 17, issue 5, pp. 694-708*, May 2008.

### Conference Publications:

1. Ai-Mei Huang and Truong Nguyen, "Motion Vector Processing Based on Trajectory Curve Analysis for Motion Compensated Frame Interpolation," submitted to *IEEE International Conference on Image Processing*, 2009.
2. Ai-Mei Huang and Truong Nguyen, "Motion Vector Processing using the Color Information," submitted to *IEEE International Conference on Image Processing*, 2009.
3. Ai-Mei Huang and Truong Nguyen, "Correlation-based Motion Vector Processing For Motion Compensated Frame Interpolation," in *IEEE International Conference on Image Processing*, pp. 1244 - 1247, Oct 2008.

4. Ai-Mei Huang and Truong Nguyen, “Motion Vector Processing Using Bidirectional Frame Difference in Motion Compensated Frame Interpolation,” in *IEEE Workshop on Mobile Video Delivery*, pp. 1 - 6, June 2008.

5. Ai-Mei Huang and Truong Nguyen, “A Novel Multi-Stage Motion Vector Processing Method for Motion Compensated Frame Interpolation,” in *IEEE International Conference on Image Processing*, pp. 389 - 392, May 2007.

6. Kai-Chieh Yang; Ai-Mei Huang; Truong Nguyen; Guest, C.C.; Das, P.K., “A New Objective Quality Metric for Frame Interpolation using in Video Compression,” in *IEEE International Conference on Image Processing*, pp. 177 - 180, Sept. 2007.

7. Ai-Mei Huang and Truong Nguyen, “A Novel Motion Compensated Frame Interpolation Based on Block-Merging and Residual Energy” in *IEEE Workshop on Multimedia Signal Processing*, pp. 395 - 398, Oct. 2006.

8. Ai-Mei Huang and Truong Nguyen, “Motion Vector Processing Based on Residual Energy Information for Motion Compensated Frame Interpolation,” in *IEEE International Conference on Image Processing*, pp. 2721 - 2724, Oct. 2006.

ABSTRACT OF THE DISSERTATION

## Motion Vector Processing in Compressed Video and Its Applications to Motion Compensated Frame Interpolation

by

Ai-Mei Huang
Doctor of Philosophy in Electrical and Computer Engineering

University of California San Diego, 2009

Professor Truong Nguyen, Chair

The objective of this thesis is to investigate algorithms that yield improved image quality for motion compensated frame interpolation or frame rate up-conversion. We address the problems of having broken edges and deformed structures in an interpolated frame by hierarchically refining motion vectors on different block sizes. The proposed novel, low complexity motion vector processing algorithm at the decoder explicitly considers the reliability of each received motion vector based on the received residual energy and motion vector correlation. By analyzing the distribution of residual energies and effectively merging blocks that have unreliable motion vectors, the structure information can be preserved.

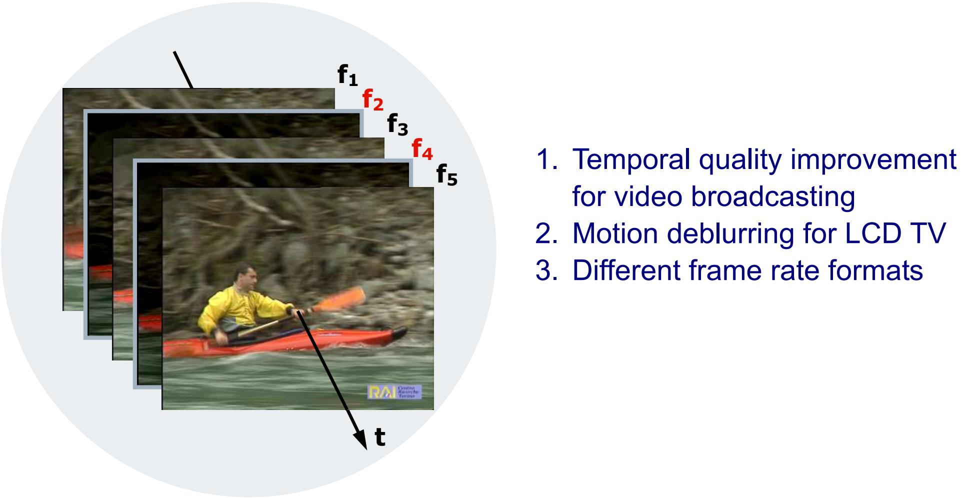
In addition to the unreliable motion vectors due to high residual energies, there are still other unreliable motion vectors that cause visual artifacts but cannot be detected by high residual energy or bidirectional prediction difference in motion compensated frame interpolation. We further propose a correlation-based motion vector processing to classify motion vector reliability and correct identified unreliable motion vectors by analyzing motion vector correlation in the neighborhood. These unreliable motion vectors are gradually corrected based on their bidirectional difference energy levels so that we can effectively discover the areas where no motion is reliable to be used, such as occlusions and deformed structures. For these areas, we further propose an adaptive frame interpolation scheme by ana-

lyzing their surrounding motion distribution and accurately choosing forward or backward predictions.

Since the proposed motion vector processing method exploits the spatial information such as residual energy and motion vector correlation, experimental results show that our interpolated results have better visual quality than other methods. However, we still can observe the flickering effects during video display especially in motion boundaries and areas having uniformly distributed texture. Therefore, to further ensure the temporal stability in these motion sensitive areas or video frames, a novel motion vector processing approach based on motion temporal reliability analysis is proposed. For each motion vector candidate, its temporal variation of absolute bidirectional prediction difference along the motion trajectory is examined and classified into several predefined curvatures that are obtained by motion reliability statistic analysis. Any motion vectors that can match one of the predefined curvatures will be considered as possibly temporal reliable motion. This algorithm is employed to improve the motion quality for the proposed motion vector processing method. As a result, the proposed method can effectively improve the motion accuracy for the bidirectional motion vector processing and outperforms other approaches in terms of visual quality, PSNR (Peak Signal to Noise Ratio), and structure similarity.

# 1 Introduction

The application of motion-compensated frame interpolation (MCFI) techniques to increase video frame rate at playback has gained significant attention in both academia and consumer electronics industries for the last decade. This is because MCFI improves temporal resolution by interpolating extra frames along the motion trajectory, based on the assumption of linear motion as shown in Fig. 1.1. It can be used to reduce motion jerkiness for video applications that have low bandwidth requirement such as wireless video broadcasting, or to remove motion blurriness for LCD-panel display by increasing the refresh rate from 60 Hz to 120 Hz or higher. Two examples for LCD-panel display applications are shown in Fig. 1.2, which are posters demonstrated in CES 2007. MCFI requires motion information between two frames, which can be either re-estimated at the decoder or retrieved directly from the received bitstreams, depending on available resources of the devices. For example, it may be preferred to use the received motion vectors (MVs) for resource-limited handheld devices, while TV applications can afford to apply a motion estimator to obtain better MVs. Unfortunately, the received MVs or re-estimated MVs simply using block matching algorithm (BMA) are often unreliable for frame interpolation in the sense that they fail to represent true motion. Directly employing these MVs usually results in unpleasant artifacts such as blockiness, ghost artifacts and deformed structures in the interpolated frames. Therefore, it is very challenging for MCFI to produce high quality, artifact-free interpolated frames without proper MV processing.



1. Temporal quality improvement for video broadcasting
2. Motion deblurring for LCD TV
3. Different frame rate formats

Figure 1.1: Motion compensated frame interpolation or frame rate up conversion.

## 1.1 Literature Survey

In the literature, a number of motion estimation algorithms performed at the decoder have been proposed for MCFI in order to obtain true motion. The general system block diagram for the decoder-only solution is illustrated in Fig. 1.3, in which motion re-estimation is performed based on the decoded frames. Along with the commonly used BMA, MVs can be estimated by further considering spatial and temporal correlations [5][6][7][8]. Haan proposed using candidate MVs from both spatial and temporal domains as the initial search points for motion estimation (3-D recursive motion estimation) so that the motion vector field (MVF) can rapidly converge to a coherent state. To further promote its efficiency, a parametric motion model was used to generate the global motion from a previously estimated MVF as candidate vectors [9]. Ha et al. suggested considering the spatial pixel information by using an overlapped block-based motion estimation to get more accurate motion trajectory [8]. The software solution for 3-D recursive motion estimation was presented in [10], in which the block sizes for motion estimation are adaptively adjusted based on spatial pixel variations. A similar concept was proposed in [2], but the block sizes for motion estimation are determined by performing a similarity measure on the previously estimated MVF. A hierarchical

Panasonic

Motion Picture Pro

Offers a 50% reduction in ghosting (MPRT basis)

clear motion images

Frame creation

120 frames/sec
Approx. 8 msec (MPRT basis)

Image interpolation between frames using motion vector detection technology

Smoother expression of motion

Dramatic reduction of ghosting to achieve clear motion images!

TOSHIBA
Leading Innovation

CLEARFRAME™

120Hz Anti-Blur Technology

A Standard LCD TV has a frame rate of 60 frames per second, or 60Hz. To avoid possible motion blur with fast motion video, ClearFrame™ doubles that frame rate to 120 frames per second, or 120Hz. While there are many possible methods for creating the additional 60 frames, ClearFrame™ utilizes superior Motion Vector Frame Interpolation technology.

By utilizing the power of the PixelPure™ processor, MVFI analyzes frame A and frame B, and creates a new frame A+B. By evaluating the speed and direction of the multiple picture elements within the frame, this

(a)

(b)

Figure 1.2: CES (2007) posters for PANASONIC MCFI technology and TOSHIBA MCFI technology.

BMA was presented in [11], where three different window sizes are used to search for true MVs based on the assumption that a large window is more suitable for finding global motions but a small window can find better local motions. To obtain more accurate predictions from forward and backward frames, instead of performing motion estimation only in one direction, Chen proposed using both forward and reverse motion estimations by exploiting the spatial motion correlation among adjacent blocks [12]. Although these motion estimation approaches can provide more reliable motion as compared to the bitstream motion, the estimated MVs still do not always represent the actual movement in the interpolated frame, especially in the motion boundary regions. As a result, additional MVF refinement from the current decoded frame to the interpolated frame is often required.

To obtain the motion of the interpolated frame directly, the method in [13]

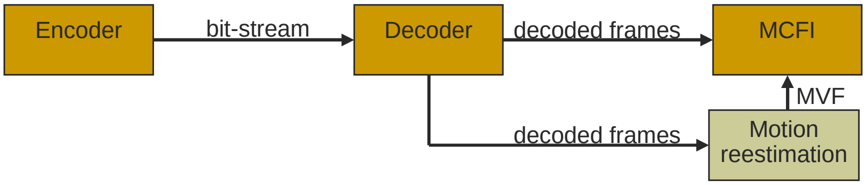


Figure 1.3: Motion re-estimation at decoder (high complexity).

performed bidirectional motion estimation and refinement for each interpolated block based on the minimal sum of bilateral absolute difference. The initial bidirectional motion search point is decided based on the results of forward motion estimation. A selectively overlapped block bidirectional motion estimation was presented in [14], in which bilateral motion estimation is only performed on the detected unreliable MVs of the received MVF. Although the motion mismatch problems for the interpolated blocks can be avoided by the bidirectional motion estimation approaches, the obtained MVF seems very unreliable when the video contains smooth background or has special motion distributions such as rotation, and zoom in/out motion. This is because MVs with minimal difference do not always represent the correct motion in these areas. Therefore, a further spatial smoothness constraint was adopted to solve the unreliable motion problem when objects have rotation or zooming motions [7][15].

In addition to these common issues of using motion estimation, the motion around motion boundaries and occlusion areas is usually very difficult to estimate accurately. Therefore, additional MV processing is often considered to improve the motion quality in these areas. By considering the motion distribution on object boundaries, image segmentation techniques are employed to further refine the estimated MVF [7][11][16][17][18]. The work in [11] tried to resolve the occlusion problems by differentiating covered and uncovered regions, but their motion processing algorithms for these areas were not mentioned. In order to reduce the sensitivity for motion estimation in occlusion areas, Lunter proposed three-frame-matching by considering both previous frame and future frame for the 3-D recursive search method, in which bad motion matches are prevented by statistic analysis [19]. To obtain more natural motion flow, Csillag *et al.* proposed using an accelerated motion model for MCFI [20].

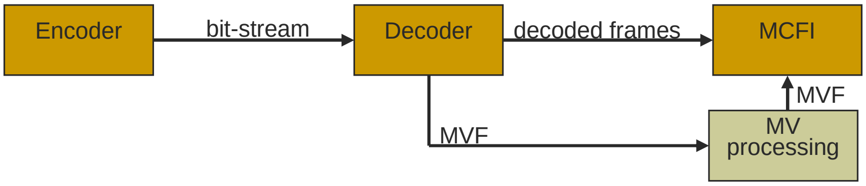


Figure 1.4: Motion vector processing at decoder (low complexity). The bitstream MVF is further porcessed for frame interpolation.

On the other hand, if motion estimation is not possible at the decoder because of higher computational complexity, motion vector processing techniques can also be used to obtain smoother motion from the received MVF by simply removing MV outliers and/or refining MVs from its neighborhood as shown in Fig. 1.4. Vector median filter (VMF) is a commonly used method to remove motion outliers [21]. An adaptive VMF was then presented in [3] by considering displaced frame difference at the encoder. Dane et al. addressed the same concept and applied VMF at the decoder to correct irregular MVs [22]. Sekiguchi *et al.* further used weighted averaging of neighboring MVs by exploiting the received prediction errors [23]. Zhang et al. proposed a method that detects isolated MVs and selects the best motion from adjacent blocks based on temporal modeling [24]. In addition to MV processing, MV reliability analysis was also addressed to help correct unreliable MVs. Sasai *et al.* determined the received MV reliability by counting and calculating the number of intra blocks, isolated MVs, and MV variance [25]. A frame that has an unreliable MVF is not used for interpolation. The works in [26] and [14] also conducted a prior MV classification and only re-estimated unreliable MVs to reduce computational complexity. As most of the MCFI approaches are block-based, blocking artifacts are easily observed in the interpolated frames. In order to provide better visual quality, overlapped block motion compensation was adopted to smooth out the blockiness along the block boundaries [13][7]. Alternatively, instead of eliminating the blockiness artifacts in the pixel domain, the works in [27] and [1] resampled the received MVF into smaller blocks and minimized the difference among these finer MVs.

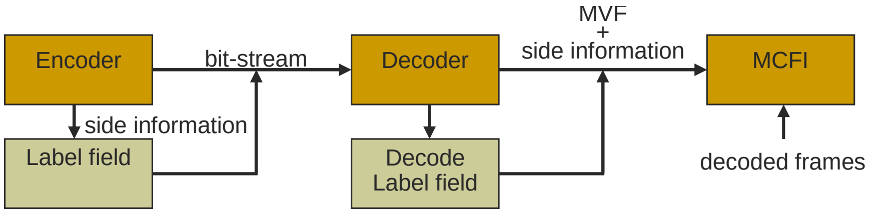
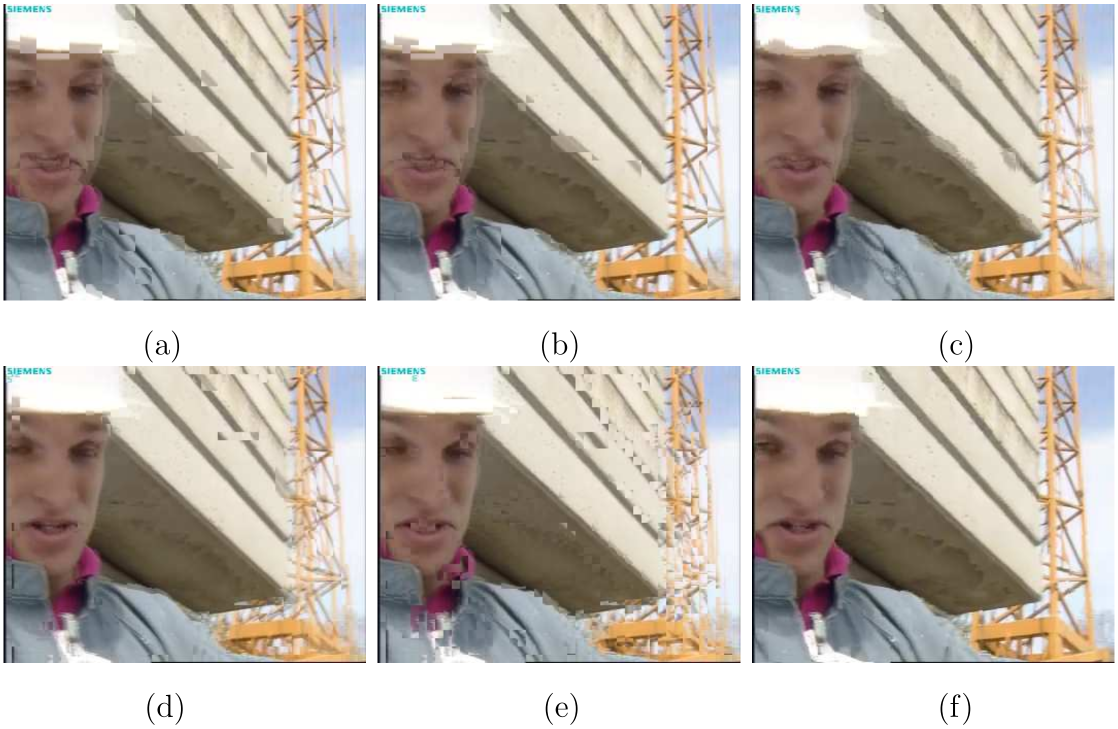


Figure 1.5: Encoder assisted (non-standard compliant). Motion estimation or image segmentation are performed at encoder, and this information is encoded and sent to decoder.

In addition to MV processing, several works have also discussed how to reduce visual artifacts by adaptively choosing forward, backward, or bidirectional interpolations. This is because in areas where occlusion happens, none of the surrounding MVs can represent those areas well. Algorithm for reducing the visual artifacts in occlusion areas was addressed in [28], where a weighted averaging interpolation by considering multiple motion trajectories and the corresponding prediction errors was proposed. A pixel-wise interpolation approach using non-linear filtering was addressed in [29]. Chen proposed adaptively choosing the forward and backward predictions based on the block boundary absolute difference [12]. Similar concepts were also presented in [6], [30], and [31], but the prediction selection was performed at the encoder and this information was sent as side information to the decoder. A theoretical analysis of adaptively choosing forward and backward interpolation was presented in [32]. The works in [11] and [17] embedded image segmentation information in the bitstream as side information so that the decoder can choose better predictions for uncovered and covered areas.

In general, it is difficult for an encoder to accurately capture all the motion in a video frame using block-based motion estimation, or coding efficiency will significantly be reduced in order to send such detailed motion information. It is also unrealistic to assume that all the encoders are made aware of frame interpolation at the decoder. Even though MVs can be re-estimated at the decoder, the true motion can easily be distorted due to coding artifacts such as blockiness and blurriness. The MV processing methods that remove outliers using VMF or refine MVs using smaller block sizes can only perform well in areas with smooth



(a) (b) (c)

(d) (e) (f)

Figure 1.6: The interpolation frames of Foreman frame 182. (a) Direct MCFI. (b) VMF. (c) MV smoothing [1]. (d) Multisize BMA [2]. (e) Adaptive VMF [3]. (f) The proposed method [4]. Here, adaptive VMF is implemented at decoder with a prior full-search motion re-estimation based on the block size of 8×8.

and regular motion. That is, they are based on the assumption that the MVF should be smooth. However, this is usually not true as a video frame may contain complex motion, especially on motion boundaries, where the true motion field is not smooth at all. As a result, irregular motion may appear in the received MVF and dominate the VMF process to use those irregular MVs as the true motion. In addition, since many of the methods only operate on a smaller block size, they often fail to consider the edge continuity and the structure of the objects. We can often see broken edges and destroyed structures in an interpolation frame. Besides, macroblocks (MBs) that are intra-coded also make frame interpolation difficult as their MVs are not available. Some methods use object-based motion estimation and interpolation at the decoder to maintain the object structure and minimize the interpolation error. However, high computational complexity may prevent them from being used in resource limited devices such as mobile phones.

Fig. 1.6 demonstrates interpolation results using conventional MCFI methods. As observed, although Fig. 1.6(b) and (c) have better visual quality compared to direct MCFI as shown in Fig. 1.6(a), VMF only performs well in motion outliers and the method in [1] removes blockiness artifacts but induces more ghost artifacts at the same time. In Fig. 1.6(d) and (e), even though MVs can be re-estimated at the decoder, additional motion vector processing is still required for the motion boundary areas and the areas with repeated patterns such as wall and crane tower areas. Therefore, frame interpolation still remains a very challenging problem as the artifacts due to the use of improper MVs can be noticeable, or an extremely complex method has to be employed.

## 1.2 The Proposed Motion Vector Processing for MCFI

To solve the aforementioned problems in MCFI, we propose a novel hierarchical, low-complexity MV processing method that can successfully preserve the edges and object structure information without involving motion estimation or object detection at the decoder. Based on the received information, we first identify MVs that are likely to produce visual artifacts during frame interpolation by exploiting the strong correlation between the reliability of a MV and the residual energy it produces. That is, residual energy of each block is analyzed to determine the reliability of the corresponding received MV. Then, before refining those unreliable MVs by further partitioning each block into smaller blocks, we propose to merge MBs that have unreliable MVs by analyzing the distribution of the residual energies. This MB merging process can effectively group MBs located on motion boundaries. In order to prevent deformed structures, each merged group is assigned a single MV selected from its own and neighboring reliable MVs by minimizing the difference between the forward and backward predictions. This is different from the method in [2], which proposes only merging global motion regions into one larger block but uses smaller blocks in local motion regions. For the new obtained MVF, we further propose an effective MV refinement method that adaptively adjusts unreliable MVs in a smaller block size (of 8×8) by applying a reliability and similarity constrained VMF to their neighboring MVs. That is, we use the reliability information of each MV as a prior knowledge in the refinement process. Unlike the method in [3], which uses prediction difference as the weighting factor for each MV in VMF, our method simply does not use those neighboring unreliable MVs. In addition, in order not to choose the same unreliable MV, we remove identical and similar MVs in the neighborhood from consideration. To further reduce blockiness artifact, we adopt MV smoothing as in [1] on an even finer block size (of 4x4) as the last step in our MV processing method. For those MBs on the frame boundaries, we propose using unidirectional interpolation by adaptively selecting forward and backward predictions based on the motion. Also, chrominance information is used in MV reliability classification and in all MV processing stages, which is found very useful to identify and correct unreliable MVs and has not explicitly been considered in the literature.

Although the results showed that the interpolation artifacts can greatly be removed, we also noticed that not all of the unreliable MVs can be detected by their residual energy and coding types. This usually happens when there are repeated patterns in the background. The encoder may choose MVs that deviate from the real motion but still have very low sum of absolute difference (SAD). In addition, we adopted the work in [1] as the last MV processing stage to smooth out the entire MVF to remove blocking artifacts, but it also created ghost artifact on the motion boundaries. Finally, our previous method did not explicitly consider the problem when occlusion happens, which may also cause ghost artifacts. Therefore, based on this discussion, we further extend our previous method to solve these problems. To detect unreliable MVs that have low residual energy, we propose classifying those MVs by calculating MV correlation in their local neighborhood. Unlike VMF that removes MV outliers one at a time and usually fails when irregular MVs occur in a cluster, we merge unreliable MVs due to low MV correlation and select a single best MV from their neighbors. In addition, based on the already-calculated MV correlation information, we present an adaptive correlation-based

MV smoothing method using absolute bidirectional prediction difference (BPD) to remove blockiness artifacts but still keep structure edges. This MV smoothing process can be considered as a weighted averaging process but further takes into account the MV correlation to adaptively determine the weights. Moreover, to minimize the ghost artifacts due to video occlusion, an adaptive frame interpolation scheme is proposed to dynamically determine to use forward or backward predictions for occlusion areas based on their surrounding MVs. In occlusion areas, the MVs obtained by minimizing the prediction difference are often unreliable once the occlusion area is larger than the block size used in motion estimation. Therefore, we propose using the neighboring *corrected* MVs outside the occlusion areas but only use either forward or backward prediction. In this way, not only can the motion distribute evenly in occlusion areas, but partially deformed areas can still be reconstructed with the object motion. Compared to the methods in [11][17][20], which used image segmentation techniques and traced back several frames to determine occlusions, our method only looks at the current frame and the bidirectional frame difference of each block and hence has lower complexity.

In the proposed correlation-based hierarchical MV processing method, we aim to remove most visual artifacts for interpolation frames based on BPD values. However, the true motion flow in occlusion areas is usually very difficult to determine merely using BPD criterion. As a result, the proposed adaptive interpolation mechanism, which selects forward and backward predictions by analyzing the local motion distribution, may select inappropriate predictions if correct motion is not available. To overcome this problem, we further propose to analyze motion temporal reliability and use this reliability classification as a posterior motion quality check for bidirectional MV selection. This can be achieved by observing the resulting BPD values along the temporal axis and their associated variations. In this way, the BPD variation models of the possibly correct motion flows can be defined and used to assist in determining the motion in occlusion areas. The motion accuracy using the proposed algorithm is improved especially in motion boundaries and other areas where the actual movement is difficult to find, such as the motion of small objects.

The remaining of this dissertation is organized as follows. The general problems for true motion vector processing and its challenges for MCFI are described in Chapter 2. In Chapter 3, we present the proposed multi-stage motion vector processing method for MCFI. The related works have been published in several conferences and journals [33][34][35][36][37][38].

Based on the proposed hierarchical scheme, a correlation-based motion vector processing and an adaptive bidirectional frame interpolation scheme are presented in Chapter 4. The associated work can also be found in [39] and [4].

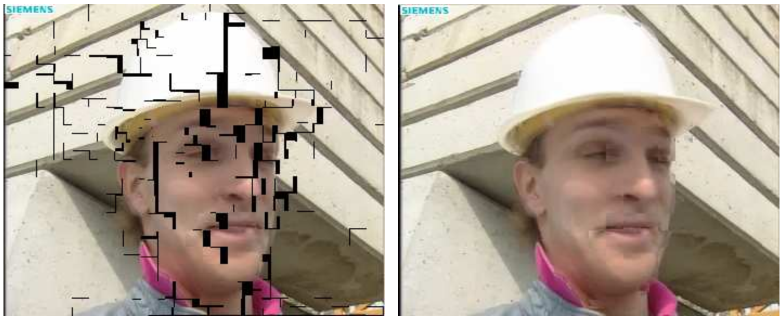
To solve the motion temporal reliability problems, the curve analysis derived from temporal BPD variations and details on how to use this curve analysis in MV processing are described in Chapter 5. The related works can be found in [40] and [41]. Finally, we summarize the contributions of the proposed research in Chapter 6.

# 2 Motion Compensated Frame Interpolation

In order to provide better temporal quality for video display, a number of methods have been proposed to improve the performance of MCFI. These include encoder-assisted MCFI, MV re-estimation, and motion vector processing based on bitstream motion, as summarized in Chapter 1. As mentioned previously, independent of the motion estimation algorithms used for MCFI, true motion can be easily distorted due to poor image quality. For those MV processing methods that correct unreliable motion by simply removing outliers, weighted averaging or MVF interpolation, it is almost impossible to recover the correct MVF from the bitstream especially for those intra-coded MBs. In this chapter, we will review two general MCFI interpolation schemes. Based on the selected scheme that is related to the proposed research, we would like to investigate its potential problems if bitstream motion is directly used in MCFI. In particular, these potential problems include 1) motion discovery for intra-coded MBs, 2) analysis and the use of color information, 3) residual energy distribution, 4) motion correlation, 5) video occlusion, and 6) temporal co-located MVs, and they are described in the following sections.

## 2.1 Backgrounds of Motion Compensated Frame Interpolation

In MCFI, the interpolated frame, $f_t$, is often obtained by one of the following two different methods:



(a) (b)

Figure 2.1: (a) Interpolated result using the direct motion trajectory from frame 3 to frame 5 of Foreman. (b) Interpolated result using the direct bidirectional MCFI from frame 3 to frame 5 of Foreman.

$$f_t(i + \frac{1}{2}v_x, j + \frac{1}{2}v_y) = \frac{1}{2} \cdot f_{t-1}(i + v_x, j + v_y) + \frac{1}{2} \cdot f_{t+1}(i, j), \tag{2.1}$$

or,

$$f_t(i, j) = \frac{1}{2} \cdot f_{t-1}(i + \frac{1}{2}v_x, j + \frac{1}{2}v_y) + \frac{1}{2} \cdot f_{t+1}(i - \frac{1}{2}v_x, j - \frac{1}{2}v_y), \tag{2.2}$$

where $f_{t-1}$ and $f_{t+1}$ denote the previous and current reconstructed frames, respectively. $\mathbf{v} = (v_x, v_y)$ is the MV used to reconstruct the frame $f_{t+1}$, or it can be re-estimated at the decoder.

The first method using Eqn. (2.1) assumes the interpolated frame can be produced by motion compensating from $f_{t-1}$ and $f_{t+1}$ along the motion trajectory. If a block-based MCFI is used, holes and overlapped regions frequently appear in interpolation results. As a result, a computationally expensive spatial interpolation is often adopted to fill holes [42]. In addition, tracking the motion trajectory and recording holes and overlapped regions can be a complicated process. Different from the complex motion-trajectory based MCFI scheme, the second method in Eqn. (2.2) simply takes the MVs of the co-located blocks and divides them by two to form forward and backward MVs. Then, the interpolated frame $f_t$ can be obtained by averaging these two predictions. This method can also be referred to as bidirectional MCFI. Although there is no spatial interpolation required, ghost

Table 2.1: Statistics of intra-coded MBs for 6 sequences

| Sequences | Average number of intra MBs | Maximum number of intra MBs |
|---|---|---|
| Foreman | 42(11%) | 311(79%) |
| Walk | 92(23%) | 304(77%) |
| Formula 1 | 97(24%) | 282(71%) |
| Fast Food | 120(30%) | 384(97%) |
| Football | 121(31%) | 336(85%) |
| Stephan | 46(12%) | 201(51%) |

artifacts can easily occur in the interpolated frame if the received MV does not represent the real motion. Examples using these two schemes are demonstrated in Fig. 2.1(a) and (b), where black areas represent holes. In Fig. 2.1(b), we can observe that ghost artifacts occur at the motion boundary or the places where MVs are unreliable. In the proposed method, we use the second scheme in Eqn. (2.2) to interpolate a frame because of its simplicity.

Fig. 2.1 can also be referred to as *direct* MCFI, if the received MVF is directly used for MCFI. The received MVs are usually computed at the encoder by maximizing coding efficiency, instead of finding true motion, so they are often unreliable and do not represent the actual motion. As observed in Fig. 2.1, the quality of the interpolated result mainly relies on the reliability of the received MVF. In the following section, we further discuss the potential problems that may cause visual artifacts in the motion compensated interpolated frames.

## 2.2 Intra-coded Macroblocks

Different from other video codecs, H.264 takes great advantage of intra mode based on previously encoded and reconstructed blocks. That is, H.264 is likely to have more intra-coded MBs compared to other standard video codecs. As a result, it becomes very difficult to recover the bitstream motion using conventional MV processing methods and a frame repeat method is usually adopted.

To demonstrate the statistics of intra mode for H.264 encoder, six video sequences of CIF size are encoded with 15 fps and bitrate at 384kps to 512kbps as shown in Table 2.1. We calculate the number of intra-coded MBs for each frame to obtain the average values and maximum values, in which the scene change frames are excluded. As observed, even the sequence with simplest motion, such as FOREMAN, still contains about 42 intra-coded MBs for each frame on average. Most intra-coded MBs occur in the areas of face. When fast panning motion starts, the number of intra-coded MBs reach the highest ratio, 79% (frame 193). Since large motion may result in the increase on coding bits, the encoder chooses intra mode to encode sky and shirt areas rather than inter mode. In FAST FOOD, although correct MVs can be found from the reference frame, H.264 still prefers using intra mode to encode the shirts and background areas. This is because these areas are either composed of smooth contents or special patterns, and intra mode seems to have higher coding efficiency than inter mode. Hence, not only will the distorted objects and occlusion regions be encoded as intra-coded MBs, but also the areas that can match one of 13 optional prediction modes will be encoded as intra-coded MBs in H.264. In such cases, simply assigning MVs to intra-coded MBs by VMF or MV averaging cannot guarantee the success of the interpolation frame.

Instead of re-estimating MVs for intra-coded MBs, we can actually propagate the correct existing motion from inter-coded MBs in the neighborhood to intra-coded MBs. According to our observation, even though MBs of the internal object may be encoded as intra mode, most MBs around motion boundaries are still encoded as inter-coded MBs. That is, suitable motion usually can be found around the object contour and can be further assigned to the MBs that are inside the object. Another reason for doing this is that as the number of MV candidates become fewer, the resulting MVF is more likely to have coherent distribution. Many conventional motion estimation approaches also adopt a similar concept of using MV candidates to refine the current MVF [5][13]. The details about the MV assignment for intra-coded MBs is described in Chapter 3.

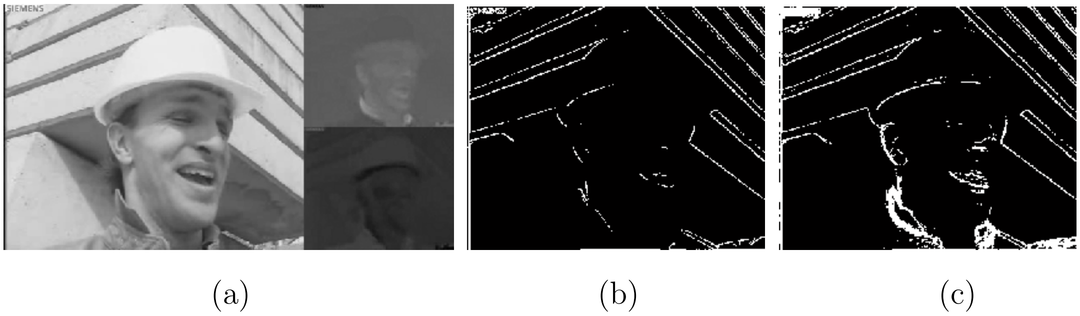


Figure 2.2: (a) The Y, Cb, and Cr components for frame 13 of FOREMAN sequence. (b) The object edge detection map using the luminance information. (c) The object edge detection map using both the luminance information and the chrominance information.

## 2.3 Color Information

Conventionally, most motion estimation approaches search the motion in the luminance domain and it is not clear whether color information has been considered in the literature. However, color information has been shown to be very effective in object edges detection due to its insensitivity to specular reflection that can prevent false edge detection as compared to luminance-based methods [43]. An affine motion estimation model using image segmentation in the color domain was proposed in [44]. To obtain better motion tracking results for the human body, Lee *et al.* employed an edge refinement for labeled regions using color information [45]. Since color has sharper and more consistent variations across object boundaries, applications that need accurate object edge information often take color information to assist the image segmentation process. Comparing to other conventional luminance-based approaches, their experimental results showed that the visual quality can benefit from using color information.

As shown in Fig. 2.2(a), in which the Y, Cb, and Cr maps are demonstrated clockwise, we can observe that luminance components often have stronger intensity distribution than chrominance components. Consequently, conventional motion estimation and MV processing approaches usually ignore color information and only use luminance information. On the other hand, color information has been widely exploited in image edge detection and image segmentation fields. This is because some color characteristics are distinct from luminance, such as the insensitivity in highlight areas. A simple edge detection experiment is demonstrated in Fig. 2.2(b) and (c). Comparing these two results, chrominance components obviously improve the edge identification for the static text, face features, the cap, and the shirt. With additional color information, not only do the object edges become sharper but the areas where luminance cannot produce clear edges are effectively detected in Fig. 2.2(c). The reason we analyze the color gradient strength around object boundaries is that if moving objects have sharp edges, ambiguous motions are more unlikely to appear. The work in [46] also considered edge information during the motion search based on the assumption that motion boundaries should align with object boundaries, but its edge information is obtained merely using luminance components. According to this edge detection result, color information seems very useful in differentiating object boundaries especially in areas where the luminance component tends to distribute uniformly. Therefore, we should further take color information into account to assist MV processing in MCFI. The details about how to use color information to refine the motion for motion boundary areas will be discussed in the next section and Chapter 3.

## 2.4 Video Occlusions

Areas where new objects appear or existing objects disappear can be referred to as video occlusions. As an object moves from $f_{t-1}$ to $f_{t+1}$, the background area that is originally covered by the object in $f_{t-1}$ becomes uncovered in $f_{t+1}$. In those areas, the encoder may decide any MV that yields smaller prediction errors, or an intra-code mode is used. The problem with covered and uncovered regions is that the MVs of the co-located blocks in the interpolation frame cannot be decided as none of the MVs in $f_{t+1}$ can be used in bidirectional MCFI. An example is shown in Fig. 2.3, in which the object should be in the middle of the motion trajectory in $f_t$. We can observe that the covered areas in $f_{t-1}$ and the uncovered areas in $f_{t+1}$ (and vice versa) become occlusions since perfect matches from $f_{t-1}$ to $f_{t+1}$ may not exist. For these occlusion areas, which are indicated by gray in

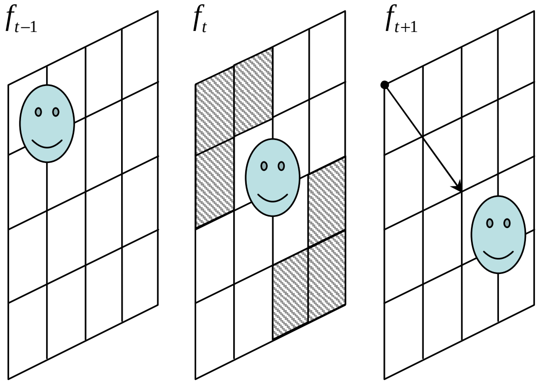


Figure 2.3: Possible occlusion areas for a moving object in the bidirectional MCFI scheme.

Fig. 2.3, the visual artifacts caused by occlusion cannot be removed completely even though we have the correct MV for the moving object. A commonly observed case is the frame boundary where it is difficult to find matched predictions if the video content is not static.

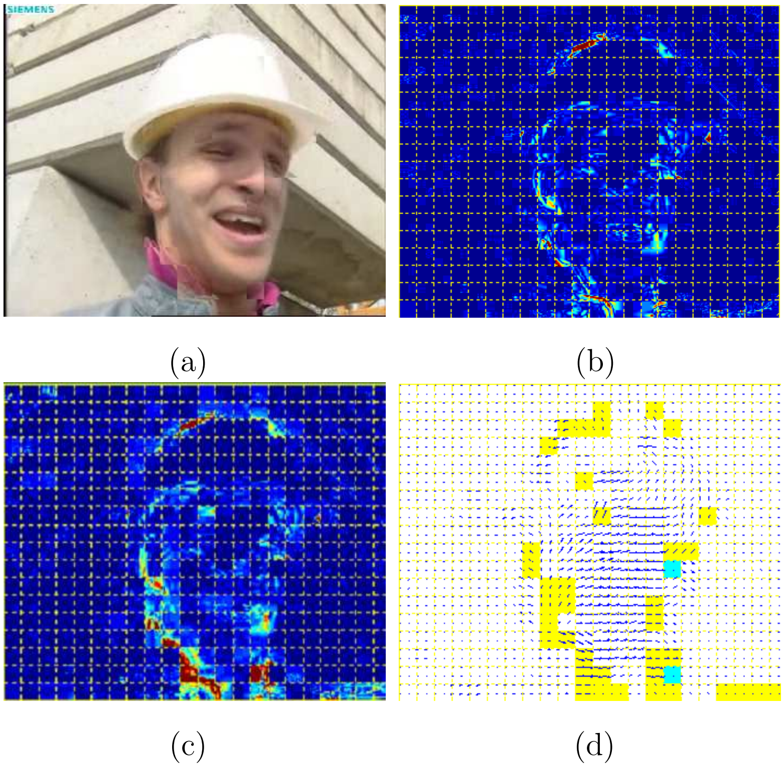
Conventionally, image segmentation techniques are often used to identify uncovered and covered regions. To reduce ghost artifacts, the work in [28] imposed blurriness effect on the motion boundary. In order to remove the artifacts on the frame boundaries, we have proposed using unidirectional frame interpolation by examining whether the MVs point to outside or inside the frame boundary in Chapter 3. In this way, we can choose better motion compensation that is located inside the frame to avoid ghost artifacts. However, the occlusion can occur anywhere in the frame but the proposed method only works well on the frame boundaries. To solve the general occlusion problem in block-based frame interpolation, the motion distribution around the occlusion region should be explicitly analyzed to determine which movements most likely cause the occlusion. That is, based on this information, the appropriate interpolation scheme that adaptively selects forward or backward prediction can be used to alleviate the ghost artifacts due to occlusion. The related research about motion correction in occlusion regions is presented in Chapter 4 and Chapter 5.

## 2.5 Motion Vector with High Residual Energy

Block-based motion estimation usually has difficulties to represent finer motion inside a block. In such a case, a video encoder usually chooses the MVs that yield the smallest prediction errors and encodes the residues to compensate for areas with different motion. Consequently, the estimated MVs become unreliable and unsuitable for frame interpolation. Visual artifacts occur frequently when those unreliable MVs are used in Eqn. 2.2. For example, Fig. 2.4(a) shows the interpolated frame 14 in FOREMAN using the direct MCFI method from reconstructed frames 13 and 15. The received MVs in frame 15 are used for interpolation. Artifacts such as blockiness and deformed structures appear in areas where the residual energies are high or in areas where no MV is available. These artifacts can also be observed even when the frame is interpolated by advanced MV processing methods such as in [21] and [22]. That is, the MVs of these regions cannot be corrected by using these conventional approaches either because the majority of the neighboring MVs belong to different motion when a median filter is used, or because the wrong MVs are used to generate a smooth motion field.

In order to illustrate our observation described above, Fig. 2.4(c) shows the residual energy of each 16×16 block of frame 15. Clearly, all the artifacts appear in the MBs where residual energies are high, such as the boundaries of the face and the ear, and the edges of the collar and the neck. From Fig. 2.4(a) and (c), we can observe that there exists a strong correlation between MV reliability and its associated residual energy. That is, when residual energy is high, it is likely that the corresponding MV is not reliable for frame interpolation. The analysis of residual energy distribution and how to use the residual information to maintain structures of moving objects are elaborated in Chapter 3.

In Fig. 2.4, we also demonstrate two different residual maps. One contains luminance residues only, and another contains both luminance and chrominance residues. As observed, since luminance components have very smooth variations around the face and shirts areas, wrong MVs can easily occur in these areas if the encoder takes the luminance difference as the major criterion to determine motion.

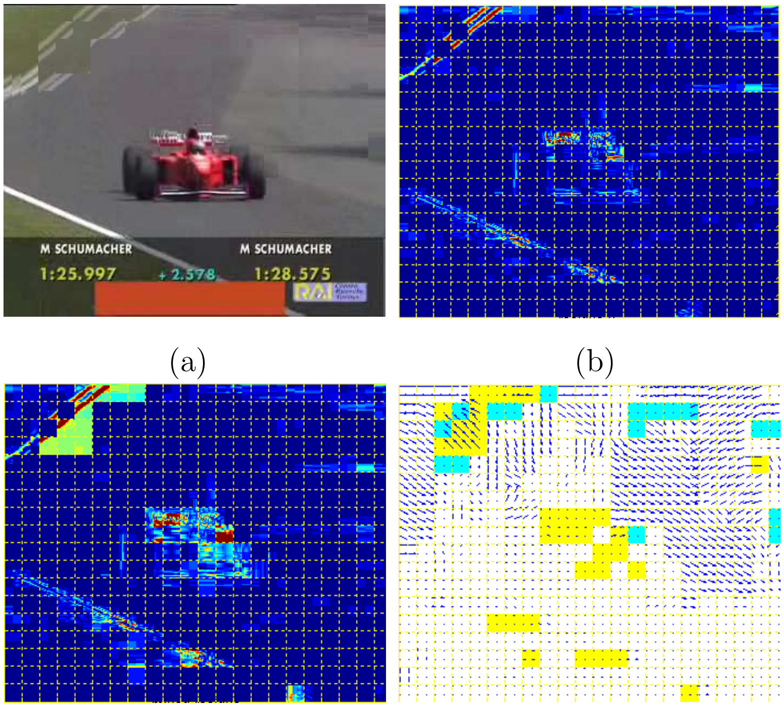


(a) (b)

(c) (d)

Figure 2.4: (a) Interpolation result of frame 14 of Foreman using direct MCFI from reconstructed frames 13 and 15. (b) Luminance residual energy map. (c) Combination map of the luminance residual energy and the chrominance residual energy. (d) MV reliability classification map for the reconstructed frame 15 based on the combination residual energy map (yellow MBs), and intra-MBs (cyan MBs).

Comparing the received residual energy in Fig. 2.4(b) and (c), the unreliable MVs around the shirt can only be detected using both luminance and chrominance information. This is because the luminance seems to have a uniform distribution in the shirt region so that the encoder always chooses the face motion for the shirt. In such cases, these unreliable MVs can only be distinguished using the corresponding chrominance residues. As shown in Fig. 2.2(c), the color distribution certainly has stronger gradients than the luminance does around the shirt. Therefore, in the proposed MV processing method, we should take advantage of the color information around the motion boundaries to identify the unreliable MVs that have small luminance residues but point to wrong objects with similar intensity. The corresponding MV reliability map using combined residual energy is presented in



(a) (b)

(c) (d)

Figure 2.5: (a) Interpolation result of frame 56 of Formula 1 using direct MCFI from reconstructed frames 55 and 57. (b) Luminance residual energy map. (c) Combination map of the luminance residual energy and the chrominance residual energy. (d) MV reliability classification map for the reconstructed frame 56 based on the combination residual energy map (yellow MBs), and intra-MBs (cyan MBs).

Fig. 2.4(d) where MBs having at least one unreliable MV are denoted by yellow. According to this map, most artifacts in Fig. 2.4(a) can be identified by analyzing the high residual energy values.

Another comparison is given in Fig. 2.5. The pavement and lawn have very similar intensity, so incorrect MVs cannot be detected merely by using the luminance residual energy as shown in Fig. 2.5(b). If we consider both luminance and chrominance information during the MV reliability classification, these unreliable MVs can be effectively identified as shown in Fig. 2.5(c) and (d). Generally, the chrominance residual distribution is similar to the luminance components but with smoother variations. However, once the encoder favors the motion belonging to different objects due to the plain luminance changes, the color information becomes relatively important compared to the luminance components. By imposing the color information in the residual energy, the accuracy of MV reliability classification can be greatly improved due to stronger gradient variations of the residual distribution.

## 2.6 Irregular Motion Vectors

Generally speaking, the MVF between two frames is supposed to be smooth, except at the motion boundaries and occlusion areas, which usually can be detected by high residual energy. However, irregular motion with low residual energy can still occur, depending on how the motion estimation is performed at the encoder. Moreover, the encoder usually takes both prediction errors and differences between adjacent MVs as coding difficulties, so once an irregular MV has been determined, it may affect the MV decision for the following blocks, especially in an area where a repeated pattern occurs. That is, irregular MVs may appear in a cluster. VMF is often regarded as an efficient way to correct motion outliers, but it only works well for isolated irregular MVs [18][3][25][30]. If many irregular MVs happen in the same area, these unreliable MVs may dominate the performance of the VMF. Furthermore, these irregular MVs often cannot be detected by residual energy or BPD.

Fig. 2.6 shows an example of failed interpolation due to irregular MVs, in whcih Fig. 2.6(a) and (b) are reconstructed frame 181 and reconstructed frame 183 of Foreman, respectively, and Fig. 2.6(d) is the interpolated result of frame 182 using direct MCFI. Since the areas of the building structure and the human face have repeated patterns and smooth contents, block based MVs can choose any similar areas wherever the minimal absolute difference occurs, instead of finding the true motion. Therefore, the motion distribution becomes very irregular at these two areas. These unreliable MVs cause severe visual artifacts such as pattern dislocation. Ha *et al.* suggested gathering statistics about the global motion from the past few frames so that the motion estimation can favor the global motion to produce a smoother MVF for those frames that have repeated patterns [8].

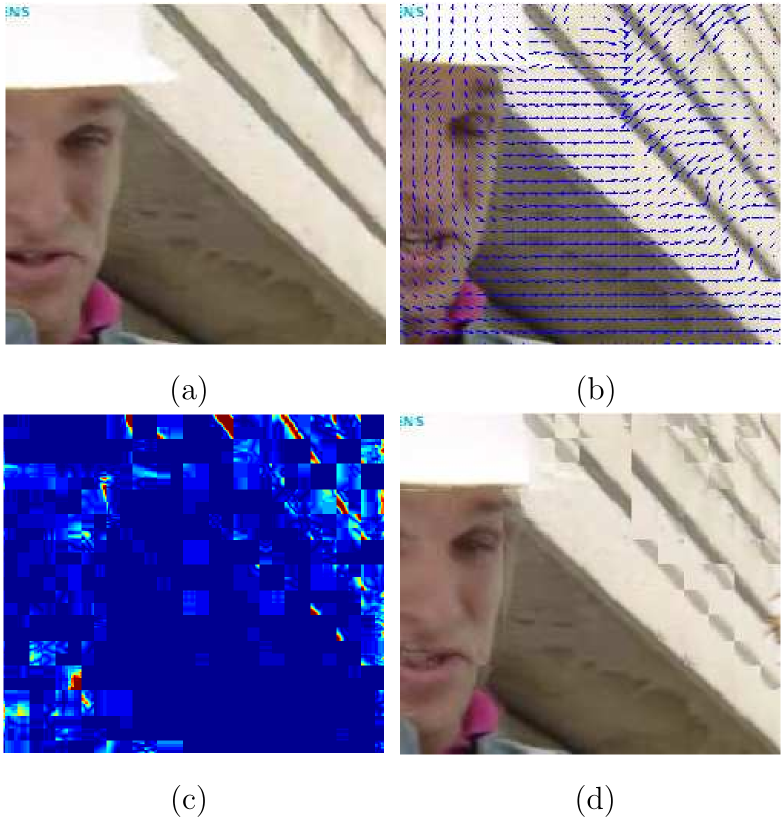


Figure 2.6: (a) The decoded frame 181 of Foreman (b) The decoded frame 183 and the received MVF (c) The received residual energy (d) Interpolation result of the frame 182 using direct MCFI from reconstructed frames 181 and 183.

However, this approach can fail easily when the video contains multiple object movement. Although most unreliable MVs can be discovered using residual energy, unfortunately, these irregular MVs often have low residual energy as shown in Fig. 2.6(c) and cannot be detected for further MV correction.

In order to effectively detect all unreliable MVs, the MV reliability classification process should explicitly consider both residual energy and MV correlation. This is because high residual energy is usually responsible for the unreliable MVs on the motion boundaries and deformed structures, while the MV correlation can reflect the motion reliability in the smooth areas and periodic scenes. In Chapter 4, the identification of unreliable MVs due to low correlation and the corresponding MV correction algorithm will be further addressed.

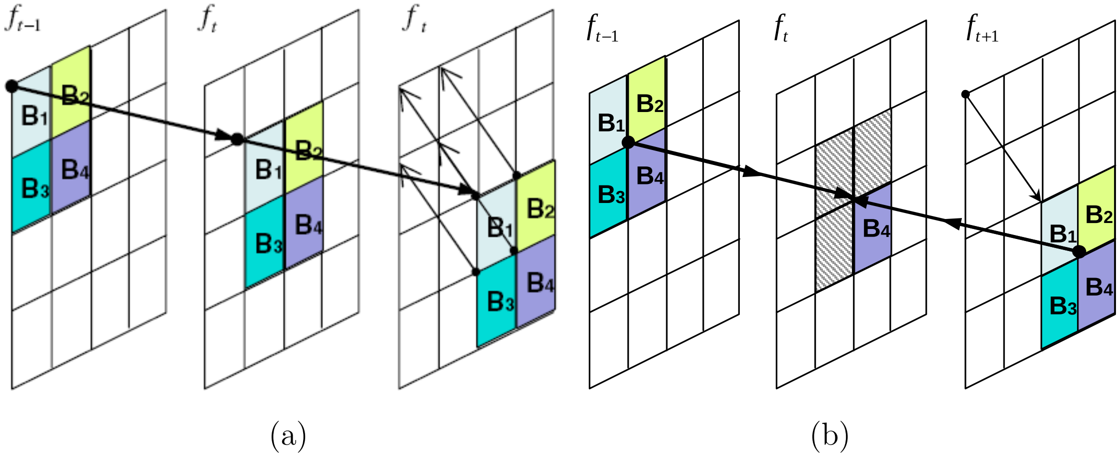


Figure 2.7: (a) Synthetical interpolation result using the motion trajectory scheme. (b) Synthetical interpolation result using the bidirectional MCFI scheme.

## 2.7 Co-located Motion Vectors

The received MVs from the bitstream are often obtained using block based motion estimation from $f_{t-1}$ to $f_{t+1}$. Even though these MVs may represent true motion for blocks in $f_{t+1}$, they may not represent the motion of the temporal co-located blocks in $f_t$. This is because the movement is not estimated based on the object position in the interpolated frame $f_t$ but based on the current reconstructed frame $f_{t+1}$. Once an object moves across more than one block from $f_{t-1}$ to $f_{t+1}$, those blocks between the object's previous position and current position may have different motion or even different coding modes. Since the bidirectional MCFI takes the MVs of the co-located blocks in $f_{t+1}$, those MVs may fail to capture the object movement in $f_t$ and result in ghost artifacts.

We use Fig. 2.7 to illustrate the problem of using the co-located MVs, in which a moving object is composed of four blocks, $\{B_1, B_2, B_3, B_4\}$. From $f_{t-1}$ to $f_{t+1}$, the object moves across two blocks in both horizontal and vertical directions. If we follow the actual motion trajectory, the correct interpolated result should look like Fig. 2.7(a). On the other hand, if the co-located MVs are employed for bidirectional MCFI, the moving object in $f_t$ cannot be interpolated successfully, denoted as shaded blocks in Fig. 2.7(b). This is because these interpolated blocks may not have the same motion as $B_1$, $B_2$, and $B_3$ in $f_{t+1}$. As a result, we can often observe ghost artifacts around the contour of the reconstructed moving objects. Even when MVs of the moving object are correctly estimated in the current decoded frame, the interpolated moving object still cannot be perfectly reconstructed. This phenomenon becomes more noticeable when video frames are skipped to reduce the bandwidth, since the temporal displacement between two consecutive frames becomes larger.

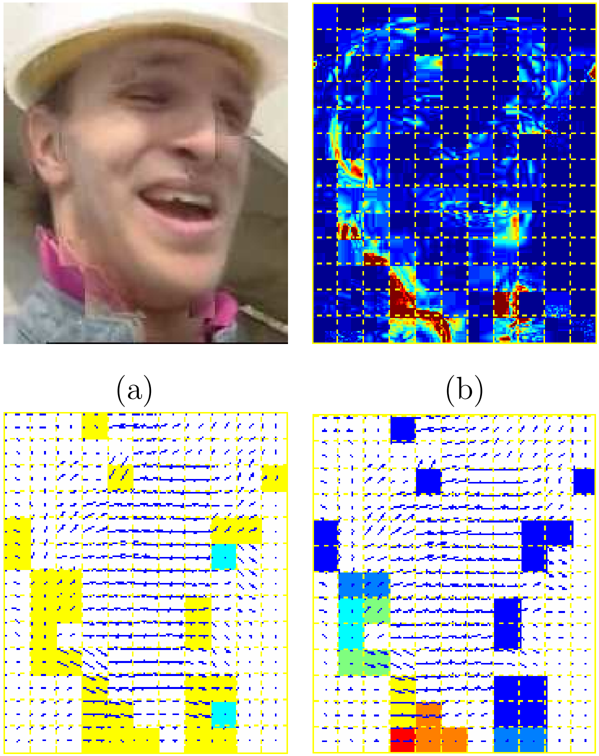
In bidirectional MCFI, not only does the received MVF have this problem, but any other block based motion estimation algorithms that only consider unidirectional motion would encounter the same difficulty. To overcome this problem, bidirectional motion estimation can be used to find the best MV based on the minimum absolute bidirectional difference, instead of merely using forward or backward motion estimation [7][13][12][14]. Although these motion estimation algorithms provide better visual quality compared to the received MVF, high complexity makes them unfeasible for actual applications on mobile devices. Some other works suggested tracing motion trajectory projection on the interpolated frame [18][23] but these approaches assume the received MVs are accurate, which is not always true. Based on this discussion, we can therefore conclude that the received MVF cannot be directly employed for the co-located interpolated blocks without further bidirectional MV correction. Since we know these ghost artifacts are produced due to unmatched forward and backward predictions, the simplest way to reduce these visual artifacts is to find the best matches by minimizing the difference between bidirectional predictions. More details will be described in Chapter 3.

## 2.8 Acknowledgement

Portions of this chapter appear in “A Multistage Motion Vector Processing Method for Motion-Compensated Frame Interpolation,” in *IEEE Transactions on Image Processing*, May 2008 and “Correlation-Based Motion Vector Processing with Adaptive Interpolation Scheme for Motion Compensated Frame Interpolation,” in *IEEE Transactions on Image Processing*, April 2009. The dissertation author was the primary author of these publications, and the listed co-author directed and supervised the research that forms the basis for this chapter.

# 3 A Multi-Stage Motion Vector Processing Method Based on Residual Energy

From the discussion on MV reliability in the previous chapter, we know that most unreliable MVs are associated with high residual energy, which can be easily detected by calculating the residual energy values. Another observation from residual energy is that the high residual energy values often occur along the object edges or cluster in the areas of deformed structures. In Fig. 3.1(a), we can observe that the structure of the shirt collar is not maintained if the received MVs are directly employed for interpolation. The encoder seems to favor the motion of the cheek, instead of the collar. As a result, these blocks are compensated with higher prediction residues as shown in Fig. 3.1(b). In most existing MCFI methods, these MVs are usually re-estimated or corrected separately. However, there is no guarantee that blocks that belong to the collar will have the same MV to perfectly assemble the collar. If we look closely at how these high residual energies are distributed, we can roughly tell the boundary where the motion starts to differ. Then, for the portions that belong to the collar, we should assign them the same MV so that the structure can be maintained before any MV refinement. The easiest way to do so is to properly merge those blocks by carefully analyzing the *connectivity* of their residual energies and find a single MV. In such way, we can avoid having disconnected structures by finding an object motion for a merged group. Therefore, we propose using residual energy to assist MV processing by



(a) (b)

(c) (d)

Figure 3.1: (a) Interpolation result of frame 14 of Foreman using direct MCFI from reconstructed frames 13 and 15. (b) Residual energy of the reconstructed frame 15. (c) MV reliability classification map. Unreliable MVs are marked in yellow and intra-coded MBs are marked in cyan. (d) MB merging map.

creating a MV reliability map and a MB merging map.

The MV reliability map is to determine the reliability level of each received MV to make sure that unreliable ones are not used for MV correction. The MB merging map is to tell whether the neighboring MBs should be grouped together in order to maintain the integrity of the entire moving object. According to the MV reliability map and the MB merging map, we propose a novel hierarchical MV processing approach in the sense that the MVs in each merged group of block size up to 32×32 are first corrected and assigned a single MV, and then these selected MVs as well as other possibly reliable MVs are further refined and smoothed based on the block sizes of 8×8 and 4×4, respectively. We first select the best MV for each merged group from its own and neighboring reliable and possibly reliable

MVs by minimizing the difference of forward and backward predictions. Please note that this object motion is still block-based in order to describe the global motion in that local neighborhood. As the selected MV can only represent the major motion within each merged group, it will be further refined based on the block size of 8×8 as part of our MV processing method to capture the detailed motion inside each merged block. This can also be seen as local motion adjustment as compared to motion selection in which a larger block size is considered to find a single MV. In the final step, we use motion smoothing technique to reduce the blockiness artifact by increasing the number of MVs based on the block size of 4×4.

In Section 3.1, we describe the algorithms to generate the MV reliability map and the MB merging map. The proposed hierarchical, multi-stage MV processing method is presented in Section 3.2. Experimental results show that the proposed method significantly improves visual quality, especially in areas with multiple motions or motion boundaries. Our method can successfully maintain object structure and has less blockiness and ghost artifacts. It is also robust even in those video sequences with complex scenes and fast motion.

## 3.1 Prediction Residual Energy Analysis and its Application for Frame Interpolation

In this section, we analyze where frame interpolation is likely to fail if the received MVs are directly used. As mentioned previously, block-based motion compensation is commonly used to reduce temporal redundancy, which is achieved by estimating MVs that minimize the prediction errors. However, when a block contains more than one motion, one single MV cannot represent all the motion in that block. Consequently, we can often observe higher residual energies on the motion boundaries or along the edges of moving objects. If those MVs are used for frame interpolation, artifacts are observed in those areas where the residual energies are high as the estimated MVs do not represent their motion. Moreover, we also observe that the high residual energy usually distributes continuously around neighboring MBs. Based on these two assumptions, we can create a MV reliability map and a MB merging map for the proposed MV processing method. The algorithms to generate the MV reliability map and the MB merging map are described below.

### 3.1.1 Motion Vector Reliability Classification

The first step is to determine if the MVs in the received bitstream are reliable. Let $\mathbf{v}_{m,n}$ denote the MV of each 8×8 block, $\mathbf{b}_{m,n}$. We classify $\mathbf{v}_{m,n}$ into three different reliability levels, *reliable*, *possibly reliable*, and *unreliable*, based on its residual energy, the reliability level of its neighboring blocks, and the coding type. The reason why we choose a block size of 8×8 for MV reliability classification is that in the MPEG-4 [47] and H.263 [48] coding standards, the prediction residues are generated and encoded based on 8×8 block size. For a MB with only one MV, we simply assign the same MV to all four 8×8 blocks.

For each block $\mathbf{b}_{m,n}$, we first calculate its residual energy, $E_{m,n}$, by taking the sum of the absolute values of each reconstructed prediction error of each pixel. In our algorithm, we consider both luminance and chrominance residues. This is because the motion estimation often uses pixel values in the luminance domain. This may result in an unreliable MV that minimizes the luminance difference but the colors are often mismatched as discussed in Section 2.3. Therefore, we include chrominance information in residual energy calculation to identify those unreliable MVs. $E_{m,n}$ can then be represented as the following:

$$E_{m,n} = \sum_{(i,j)\in\mathbf{b}^{Y}_{m,n}} |r_Y(i,j)| + \alpha \cdot ( \sum_{(i,j)\in\mathbf{b}^{Cb}_{m,n}} |r_{Cb}(i,j)| + \sum_{(i,j)\in\mathbf{b}^{Cr}_{m,n}} |r_{Cr}(i,j)|) \tag{3.1}$$

where $r_Y(i,j)$, $r_{Cb}(i,j)$, and $r_{Cr}(i,j)$ are the reconstructed residual signals of Y, Cb and Cr components of the block, $\mathbf{b}_{m,n}$, respectively. $\alpha$ is the weight used to emphasize the degree of color difference. For the selection of $\alpha$, we only need to be careful not to overemphasize the color since the luminance is still the fundamental element of the image. By observing what is the minimal required degree for the color factor so that the weighted color energy can effectively detect the unreliable MVs having small luminance difference, we empirically set $\alpha$ to be eight for 420 YUV planar clips in our experiment. The statistical analysis for selecting $\alpha$ is described in Section 3.3. Please note that the residual signals have to be reconstructed during decoding; therefore, there is no additional computation for using such information other than Eqn. (3.1).

We then compare $E_{m,n}$ with a predefined threshold, $\varepsilon_1$, to determine if $\mathbf{v}_{m,n}$ is unreliable. If $E_{m,n}$ is greater than or equal to $\varepsilon_1$, it will be considered as *unreliable* and included into the reliability set $L_1$. For intra-coded MBs, since they do not have MVs, we temporarily assign zero MVs and consider them as unreliable and place them in $L_1$.

Once an unreliable MV is identified, its neighboring MVs in the same MB and in its eight adjacent MBs will be classified as *possibly reliable* and be placed into the second reliability set $L_2$, even if their residual energy levels are below the threshold. The reason is that when one MB contains at least one block with high residual energy, this MB and the surrounding MBs are on the object edges. If these MBs are also on the motion boundary, those MVs may not represent the actual motion, depending on how motion estimation is performed at the encoder. Thus, in order to ensure that all MVs used for frame interpolation are reliable, we mark these MVs as possibly reliable and they will be revisited in a later stage of the MV correction process for further verification. For example, for a MB with four MVs, if only one block exceeds the threshold, the other three blocks as well as all the MVs in the eight adjacent MBs will be considered to be possibly reliable and put into $L_2$. But if their residual energies are high, they will still be classified into $L_1$ instead of $L_2$.

For those MVs that are not classified yet and their $E_{m,n}$ are less than $\varepsilon_1$, they will be classified as *reliable* and placed into the third reliability set $L_3$. Therefore, we can create a MV reliability map (MVRM) by assigning the reliability level to each MV as follows:

$$MVRM(m,n) = \begin{cases} L_1, & \text{if } E_{m,n} \geq \varepsilon_1, \\ L_2, & \text{if any MV in the same MB or} \\ & \text{in the adjacent MBs} \in L_1, \\ L_3, & \text{otherwise.} \end{cases} \tag{3.2}$$

Fig. 3.1(c) demonstrates the MV reliability map based on Fig. 3.1(b) using predefined values, $\alpha$ and $\varepsilon_1$. Since the luminance values of the collar and skin are very similar, some of the wrong MVs can only be detected by chrominance residues instead of luminance residues. The MVs in white are reliable MVs and those in yellow contain at least one unreliable MV. We purposely mark intra-coded MBs in cyan so that we can differentiate their impact on the interpolation quality from inter-coded MBs. However, in our reliability classification, they are considered unreliable as we initially assign zero MVs. As expected, we can successfully identify the regions where frame interpolation is most likely to fail by classifying the MV reliability.

Through abundant experiments, we found out that the proposed MV processing method is not very sensitive to $\varepsilon_1$ for many different video sequences, which will be explained in Section 3.3. Therefore, we apply the same threshold value to all video sequences in our simulation.

### 3.1.2 Macroblock Merging Based on Motion Vector Reliability

After classifying the reliability of each MV, instead of correcting those unreliable MVs separately, we should consider to merge them by analyzing the connectivity of the residual energies. The merging process is performed on a MB basis, and all MBs that contain unreliable MVs will be examined in a raster scan order. For an inter-coded MB that has unreliable MVs, we check if its unreliable MVs connect to other unreliable MVs in the adjacent MBs that have not yet been merged. That is, only those MBs that have unreliable MVs connecting to each other in vertical, horizontal and diagonal directions will be merged. If two ad-

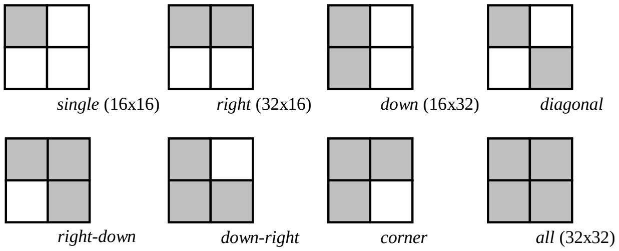


Figure 3.2: Merging shapes for inter-coded MBs that contain at least one unreliable MV and also for intra-coded MBs except for the diagonal shape.

jacent MBs have unreliable MVs that are not next to each other in those three directions, those two MBs will not be merged. If there are no unreliable MVs in the neighborhood, this MB will remain as a single 16×16 block.

All possible shapes after MB merging are shown in Fig. 3.2. All the MBs that are merged together will be given a single MV in the first stage of the proposed MV processing. We choose 32×32 as the maximum block size for CIF sequences after merging. This is sufficient to obtain a good MV to describe the object motion to maintain the edges of an object on those MBs. Further increasing the block size to 48×48 or even larger is found to reduce the quality of the interpolated frame as it is too large to represent all the motion inside. For video sequences with larger sizes, downsampling techniques are used to adjust the frame size to fit the proposed MV processing method. Moreover, the proposed method corrects MVs in a larger shape first and then refines them in smaller blocks later. Increasing the size of the merged block makes the motion refinement process difficult.

It is noted that intra-coded MBs are automatically considered in this merging process as their MVs are unreliable. We assume that the intra-coded MBs adjacent to unreliable MVs have higher prediction errors so that the encoder decides to encode those MBs using the intra-coded mode. In addition, if there are adjacent intra-coded MBs, these MBs are assumed to cover the same object in our method. That is, in the merging process, we have four types of MB merging: inter-inter, inter-intra, intra-inter, and intra-intra. However, the diagonal shape in Fig. 3.2 is not considered for intra-intra MB merging. It is because the possibility of two diagonal intra-coded MBs belonging to the same object is lower. Therefore,

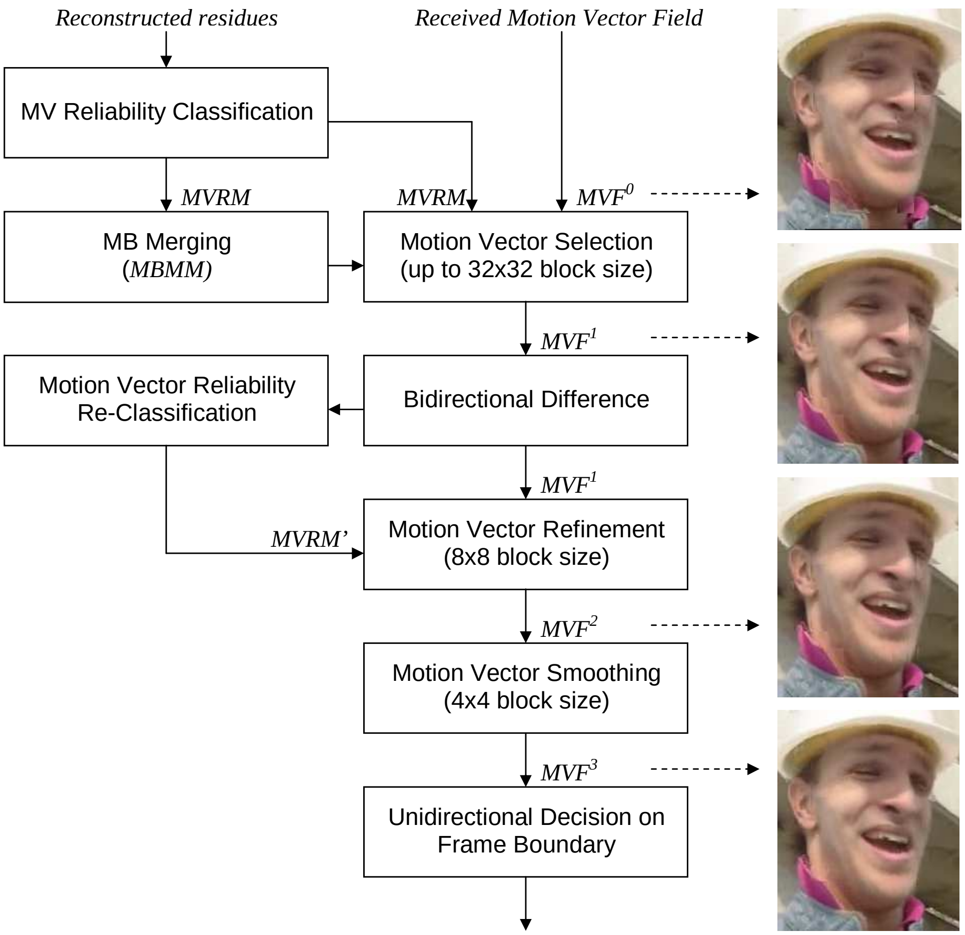


Figure 3.3: Block diagram of the proposed algorithms. $MVF^k$ denotes the updated MVF after each process.

there are 7 merging modes for intra-intra MB merging type and 8 merging modes for other MB merging types. We can create a MB merging map (*MBMM*) by assigning a unique number to the MBs that are merged, indicating that they should be considered together to find one MV in the MV processing stage.

Fig. 3.1(d) shows the MB merging map, where all the MBs in the same merged group are marked in the same color. Different colors are used to differentiate adjacent merged groups. Blue is the default color if a merged group has no other merged groups next to it. Comparing Fig. 3.1(b) with Fig. 3.1(d), we observe that some blocks with high residual energies have been grouped together to form larger blocks such as those in the collar and ear areas.

## 3.2 The Proposed Multi-stage Motion Vector Processing Method

In Fig. 3.3, according to the MV reliability map ($MVRM$), the MB merging map ($MBMM$), and the originally received motion vector field ($MVF^0$), the first MV processing step is to select the best MV for each merged group. The merged MBs will be assigned the selected MV that is used to update $MVF^0$ to $MVF^1$. Meanwhile, we check the difference of the forward and backward predictions resulted from the selected MV in the subsequent MV reclassification process. This MV reliability reclassification can help subsequent MV processing stages to differentiate improper motion as the MVs have been changed and the original residual energy is no longer useful. We refine those unreliable MVs from the second classification based on the block size of 8×8, and those corrected MVs will be used to update $MVF^1$ to $MVF^2$. Finally, we use motion smoothing to reduce the blockiness artifact by increasing the number of MVs based on the block size of 4×4. The denser MVF is denoted by $MVF^3$. In addition to the MV processing, we also adopt a different interpolation strategy for those MBs on the frame boundaries. In the following sections, we describe the proposed hierarchical, multi-stage MV processing method in greater detail.

### 3.2.1 Motion Vector Selection

The main purpose of using MB merging to correct unreliable MVs is to preserve edge information and maintain the integrity of moving object structure by finding a single MV. Instead of re-estimating motion for each merged group, we propose MV selection. From the MB merging map, the MBs in each merged group have their own MVs and also the neighboring MVs in the adjacent MBs. These MVs are the candidates for our MV selection process. That is, we choose the best MV, $\mathbf{v}_b^*$, from these candidates by minimizing the averaged absolute bidirectional prediction difference (ABPD) between the forward and backward predictions.

$$\mathbf{v}_b^* = \arg\min_{\mathbf{v}\in S}(ABPD(\mathbf{v})), \tag{3.3}$$

where

$$ABPD(\mathbf{v}) = \frac{1}{N_G}\sum_{i,j\in G}|f_{t-1}(i+\frac{1}{2}v_x, j+\frac{1}{2}v_y) - f_{t+1}(i-\frac{1}{2}v_x, j-\frac{1}{2}v_y)|,$$

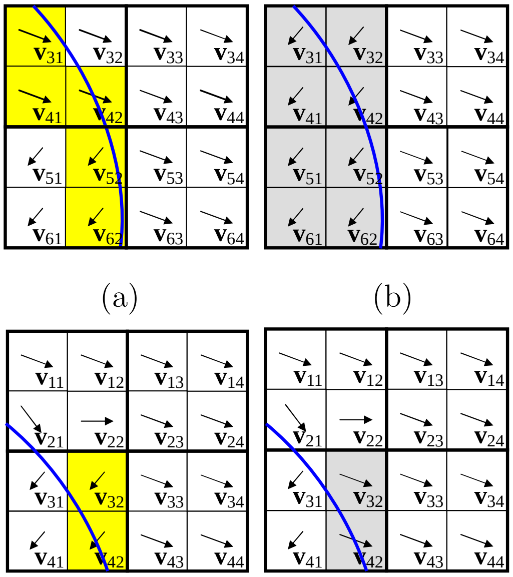
$S$ denotes the set of the MV candidates. $G$ denotes the merged group in one of the 8 possible shapes in Fig. 3.2. It is noted that we consider both luminance and chrominance information in $ABPD$ calculation in Eqn. (3.3) and use the same weighting factor as in Eqn. (3.1).

Once the best MV is found, before we assign it to the merged MBs in $G$, we need to check if this selected MV is good enough by comparing its ABPD with a threshold $\varepsilon_2$. If it is less than $\varepsilon_2$, the MVs of the merged MBs in $G$ will be replaced by the new MV $\mathbf{v}_b^*$ and marked done. However, if it is larger than or equal to $\varepsilon_2$, we drop the selected MV and skip this merged group temporarily to see if some of the neighboring MVs are updated to better ones when other merged groups are corrected. That is, we wait until a proper MV propagates to its neighborhood. If the ABPD of the selected MV is still higher than $\varepsilon_2$ and the neighboring MVs are no longer updated, we still assign the best MV $\mathbf{v}_b^*$ and refine it in the MV refinement stage. The MV selection process stops when all merged groups have been assigned new MVs, or a predefined number of iterations is reached. In our simulation, we set the iteration number to be 2 and increase $\varepsilon_2$ to a very high value in the second iteration so that all merged blocks will certainly be assigned new MVs. In Section 3.3, the process to find the best value for $\varepsilon_2$ is presented.

To further illustrate how the proposed MV selection works, an example is shown in Fig. 3.4. Each MB has 4 MVs, which are denoted as $\mathbf{v}_{mn}$, where $m$ and $n$ are the row and the column index, respectively. Assume that we have a moving object whose motion boundary is represented by the blue line as shown in Fig. 3.4(a). The left side of blue line is the moving object, and the opposite side is the background. Since the motion along the object edge differs, there should be high residual energies around the blue line. Using our MV classification approach, high residual energy areas are identified and indicated in yellow. The proposed merging algorithm groups the left two MBs, and a proper motion is assigned using Eqn. (3.3). Fig. 3.4(b) demonstrates the updated MVF, in which modified MVs are in grey. As observed, the integrity of the object structure is maintained by assigning the major motion. However, as described previously, this updated MV is still block-based and is limited in representing finer local details, such as $\mathbf{v}_{32}$ and $\mathbf{v}_{42}$, where the background motion should be chosen. Therefore, we need to perform MV reclassification and motion refinement to further correct these unreliable MVs.

### 3.2.2 Motion Vector Reclassification Based on Bidirectional Prediction Difference

After the MV selection stage, the new selected MV for each merged group is considered as a ”global” MV in that merged group. However, there may still be smaller areas inside the MBs where this new MV cannot represent their motion well. Since the residual energy can no longer provide information about this new MV, we propose using BPD energy to reclassify those MVs that are found in the MV selection stage, $MVF^1$. In addition, we revisit those possibly reliable MVs and check their ABPD to see if they are truly reliable. For those MVs that are reliable, they will still remain reliable in this stage. That is, in the MV reclassification, we only work on those MVs that are unreliable ($L_1$) and possibly reliable ($L_2$) in the first place.


$\mathbf{v}_{31}$ $\mathbf{v}_{32}$ $\mathbf{v}_{33}$ $\mathbf{v}_{34}$
$\mathbf{v}_{41}$ $\mathbf{v}_{42}$ $\mathbf{v}_{43}$ $\mathbf{v}_{44}$
$\mathbf{v}_{51}$ $\mathbf{v}_{52}$ $\mathbf{v}_{53}$ $\mathbf{v}_{54}$
$\mathbf{v}_{61}$ $\mathbf{v}_{62}$ $\mathbf{v}_{63}$ $\mathbf{v}_{64}$
$\mathbf{v}_{11}$ $\mathbf{v}_{12}$ $\mathbf{v}_{13}$ $\mathbf{v}_{14}$
$\mathbf{v}_{21}$ $\mathbf{v}_{22}$ $\mathbf{v}_{23}$ $\mathbf{v}_{24}$

(a) (b)

(c) (d)

Figure 3.4: (a) The MVF before the merging process. (b) The MVF after the merging process and motion selection. (c) The reclassification map for motion refinement. (d) The MVF after motion refinement.

The MV reclassification process is similar to the classification method described previously. Since we already use different weights on luminance and chrominance while calculating residual energy and MV selection as well, here we simply sum up difference error based on 8x8 block size using the same criteria to obtain the new energy distribution $BPD_{m,n}$.

$$BPD_{m,n} = BPD^{Y}_{m,n} + \alpha \cdot (BPD^{Cb}_{m,n} + BPD^{Cr}_{m,n}), \tag{3.4}$$

where $BPD^{Y}{}_{m,n}$, $BPD^{Cb}{}_{m,n}$, and $BPD^{Cr}{}_{m,n}$ are the sum of absolute bidirectional prediction difference for Y, Cb, and Cr components of block $\mathbf{b}_{m,n}$ using the updated MV $\mathbf{v}^{*}_{m,n}$, respectively.

If the $BPD_{m,n}$ is higher than a threshold $\varepsilon_3$, which will then the MV $\mathbf{v}^{*}_{m,n}$ will be classified as unreliable and put in $L_1$. Those MVs with $BPD_{m,n}$ lower than $\varepsilon_3$ will be classified as reliable in $L_3$. The classification can be written as

$$MVRM(m,n) = \begin{cases} L_1, & \text{if } BPD(m,n) \geq \varepsilon_3, \\ L_3, & \text{if } BPD(m,n) < \varepsilon_3. \end{cases} \tag{3.5}$$

Please note that there are no more possibly reliable MVs ($L_2$) in MVRM. All the $L_2$ MVs will be classified into $L_1$ or $L_3$ in this stage. $L_3$ MVs will remain $L_3$ in the updated MVRM. The selection process of $\varepsilon_3$ is discussed in Section 3.3.

From our observation, the difference between the forward and backward predictions usually has larger scale values than the received prediction error. As a result, we increase the threshold value to find improper MVs that will lead to noticeable artifacts. The updated classification map, $MVRM$, is the reference map in the MV refinement process. In the proposed refinement process, the scale of threshold value is not sensitive to the performance of the proposed MV processing method but we have to ensure that all significant visible artifacts can be detected effectively. As to our simulation, we use the same threshold value for all test video sequences. A simplified example is demonstrated in Fig. 3.4(c), where yellow blocks indicate unreliable MVs after the MV reliability reclassification. The reason why $\mathbf{v}_{31}$ is not classified as unreliable is that the high threshold value can only detect significant artifacts. We will leave minor difference errors for the last stage of MV processing with a finer block size.

### 3.2.3 Motion Vector Refinement

In consequence of the MV selection process, a MB in a high residual area will only have one single MV that presents major motion. As the MB consists of multiple motion, regions having different motion can be easily detected by high difference error between forward and backward predictions. Therefore, these unsuitable MVs can then be identified and reclassified in the $MVRM$.

For those unreliable MVs in $L_1$, we correct them by using a *reliability and similarity constrained vector median filter* in the following:

$$v^*_{m,n} = \arg\min_{v \in S} \sum_{i=m-1}^{m+1} \sum_{j=n-1}^{n+1} w_{i,j} \, \|v - v_{i,j}\|, \tag{3.6}$$

where

$$w_{i,j} = \begin{cases} 0, \; if \; MVRM(i,j) = L_1, \\ 1, \; if \; MVRM(i,j) = L_3 \text{ and } d_{i,j} > \varepsilon_4. \end{cases}$$

$S$ contains the neighboring MVs centered at $\mathbf{v}_{m,n}$, and $d_{i,j}$ denotes the distance between $\mathbf{v}_{i,j}$ and $\mathbf{v}_{m,n}$ using the angular difference.

$$d_{i,j} = 1 - \frac{\mathbf{v}_{m,n} \cdot \mathbf{v}_{i,j}}{|\mathbf{v}_{m,n}||\mathbf{v}_{i,j}|} = 1 - \cos\theta$$

where $\theta$ is the angle between $\mathbf{v}_{i,j}$ and $\mathbf{v}_{m,n}$. The distance is used for measuring the similarity of the candidate MVs and the original MV. Two MVs are considered to be similar if the distance is below a threshold, $\varepsilon_4$. The statistical analysis for selecting $\varepsilon_4$ is described in Section 3.3. Since we know those 8×8 blocks have different motion or belong to another object, we should try to avoid getting the same or similar MV. Hence, the proposed VMF sorts the candidate MVs that have passed the similarity check and chooses the most probable one.

Before we update the MV, we need to perform an energy check on the bidirectional difference error of the candidate MV whose error energy must be smaller than the original one. If the candidate MV $\mathbf{v}^*_{m,n}$ fails to pass the energy check, we will not update its unreliability level and try to correct it in the next iteration if a different MV can be found with an updated MVF. Similarly, we also set an iteration number to be 2 for the motion refinement process. It is possible that MVs with reliability level $L_1$ remain unreliable after the refinement. Depending on how structure information distributes on an 8×8 block, the energy check decides if the candidate MV can represent the major motion. If not, we skip this refinement and further modify this unreliable MV with finer block size 4×4 in the MV smoothing process.

We again use Fig. 3.4 to illustrate the MV refinement process. From Fig. 3.4(c) to Fig. 3.4(d), we observe that unreliable MVs of 8×8 identified by the reclassification process are effectively corrected using the proposed refinement method. Due to the similarity constraint, we prevent using identical and similar MVs to correct $\mathbf{v}_{32}$ and $\mathbf{v}_{42}$. Obviously, their new obtained MV should pass the energy check since it can better represent the local detailed motion. As we can see, high bidirectional difference may occur in finer areas in Fig. 3.4(d) and this will cause blockiness artifact. Therefore, MV smoothing is used in the last stage of the MV processing method to reduce blockiness.

### 3.2.4 Motion Vector Smoothing

In order to reduce the blocking effect, we adopt the method in [1] as our final stage of the MV processing to create a motion field with a finer scale. In [1], each 8×8 block can be further partitioned into four 4×4 sub-blocks and the MVs of these four sub-blocks can be obtained simultaneously by minimizing a smoothness measure $\Psi$, which is defined as follows:

$$\Psi = \Psi_N + \Psi_S + \Psi_E + \Psi_W + \Psi_D + \Psi_C. \tag{3.7}$$

The subscripts $\{N, S, E, W, D, C\}$ of $\Psi$ individually represents the smoothness measures between the centered MVs and their adjacent MVs in north, south, east, west, diagonal and center directions. For example, the smoothness measure of these four MVs in their north direction is defined as

$$\Psi_N \;=\; \left\|\mathbf{v}^1_{m,n} - \mathbf{v}^3_{m,n-1}\right\|^2 + \left\|\mathbf{v}^2_{m,n} - \mathbf{v}^4_{m,n-1}\right\|^2 + \left\|\mathbf{v}^3_{m,n} - \mathbf{v}^1_{m,n}\right\|^2 + \left\|\mathbf{v}^4_{m,n} - \mathbf{v}^2_{m,n}\right\|^2$$

In the equation, the MV $\mathbf{v}_{m,n}$ in the block $b_{m,n}$ is partitioned into four sub-blocks $b^i_{m,n}, i = 1, 2, 3, 4$ in scan order, with initial MV $\mathbf{v}^i_{m,n} = \mathbf{v}_{m,n}$. Similarly, we can derive smoothness measures for all other directions. The optimal solution is obtained by combining different direction smoothness measures into a matrix form and minimizing $\Psi$ in Eqn. (3.7) with respect to the four MVs.

We only use this resampling approach on MVs with original reliability levels, $L_1$ and $L_2$, because they are the major cause of visual artifacts in the frame interpolation. Please note that we use corrected MVs in $MVF^3$ and produce a denser MVF ($MVF^4$) during the smoothing process while the method in [1] uses the original received MVF ($MVF^0$). As a matter of fact, smoothness measurement aims to lessen the difference among MVs, so using proper MVs can effectively decrease blockiness whereas improper motion can induce serious ghost artifact.

### 3.2.5 Motion Adaptive Unidirectional Interpolation on the Frame Boundary

MPEG4 and H.263+ allow motion estimation to search out of frame boundary by extending the boundary pixel values for better coding efficiency. However,

Table 3.1: Weight values for forward and backward motion compensation on frame boundary.

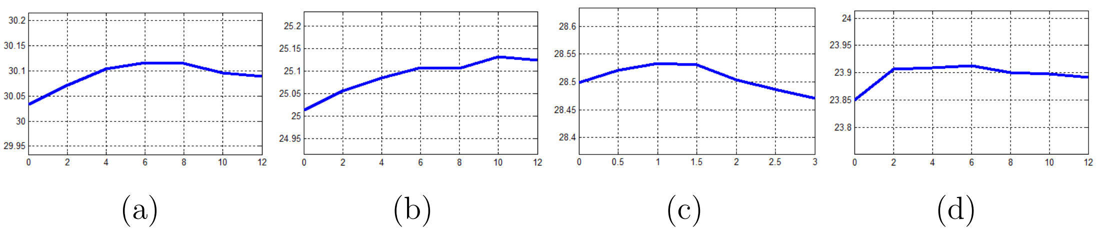
| $n = 1$ | $w_f$ | $w_b$ | $m = 1$ | $w_f$ | $w_b$ |
|---|---|---|---|---|---|
| $v_y \leq 0$ | 0 | 1 | $v_x \leq 0$ | 0 | 1 |
| $v_y > 0$ | 1 | 0 | $v_x > 0$ | 1 | 0 |
| $n = N$ | $w_f$ | $w_b$ | $m = M$ | $w_f$ | $w_b$ |
| $v_y \leq 0$ | 1 | 0 | $v_x \leq 0$ | 1 | 0 |
| $v_y > 0$ | 0 | 1 | $v_x > 0$ | 0 | 1 |

for frame interpolation, it is difficult to get good interpolation results by using bidirectional interpolation in Eqn. (2.2). For example, for MBs on the first row, if the MV in the vertical direction, $v_y$, is less than zero, it implies that a new object appears in the next frame and the previous frame only has part of th e content. Simply averaging the forward and backward predictions will cause visual artifacts. Hence, for those MBs on the frame boundary, we propose using unidirectional interpolation based on the directions of their MVs. That is, we adaptively change the weights of forward and backward predictions based on the MVs. This is summarized in Table 3.1.

## 3.3 Parameter Analysis

In our implementation, MV reliability classification, MV selection, MV reclassification, MV refinement and even the iteration limitation number all make use of predefined threshold values for the proposed MV processing method. In this section, these parameters are analyzed and their impact on the PSNR performance is also discussed in more detail. By doing this, not only can we realize how sensitive the proposed method will be to these parameters, but also we can select better thresholds and apply the same values for all test video sequences to simplify the implementation. Video sequences of CIF size is used here for the major parameter analysis, which includes $\alpha$, $\varepsilon_1$, $\varepsilon_2$, $\varepsilon_3$, $\varepsilon_4$, and iteration numbers for MV selection and MV refinement.

The iteration number is determined by comparing the input MVF and the resulting MVF. If the resulting MVF is static, MV refinement will simply stop



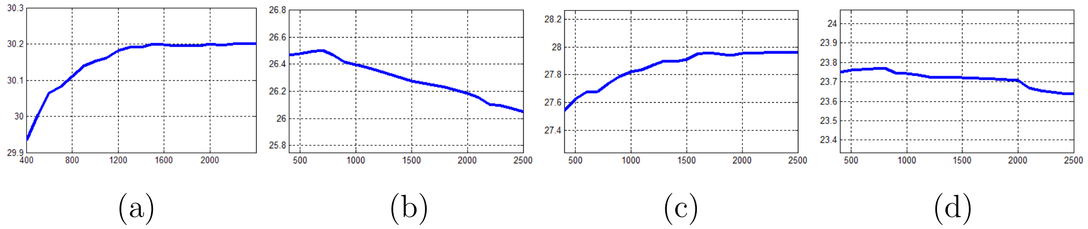
(a) (b) (c) (d)

Figure 3.5: According to different color weights used in the proposed MV processing method, (a), (b), (c), and (d) are PSNR performances for Foreman, Fast Food, Formula 1, and Stephan, respectively.

MV processing. However, MV selection will continue MV processing once again to ensure that all merged groups have been assigned a single MV when the resulting MVF no longer changes. We compute the iteration number required for each frame and obtain average values, 1.47 and 1.30, for MV selection and MV refinement, respectively. As a result, the iteration varies between 1 and 2. In order to reduce the complexity, we set the iteration number to be 2 for MV selection and increase $\varepsilon_2$ to a very high value in the second iteration so that all merged blocks will certainly be assigned new MVs. With the same manner, we also set an iteration number to be 2 for the motion refinement process.

The PSNR performance with different color weights is presented in Fig. 3.5. We can observe that as the color weight increases, the corresponding PSNR performance increases as well. However, as the color weight continues increasing, the corresponding PSNR performance degrades instead. As shown in Fig. 3.5(c), $\alpha = 12$ even performs worse than $\alpha = 0$. Since the luminance is still the major component for a image, the overall performance does not benefit by overemphasizing the importance of color components. In Fig. 3.5, Foreman, Fast Food, Formula 1, and Stephan have the best PSNR performance when $\alpha$ are 8, 10, 6, and 6, respectively. For implementation consideration, we simply set $\alpha$ to be 8 so that a bit-shift operation can be used to replace a multiplier.

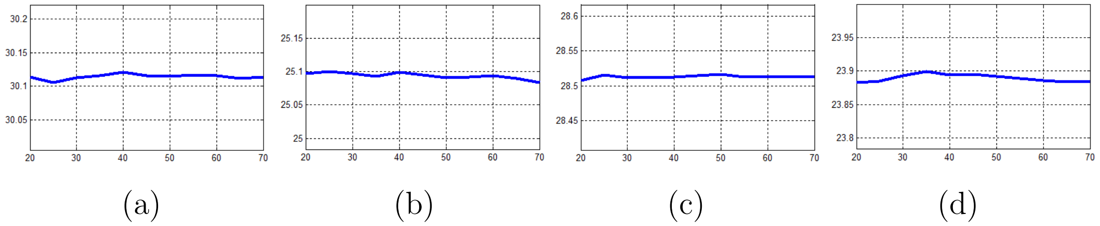
From Fig. 3.6(a) to (b), the corresponding PSNR performance comparison with different residual energy threshold values is presented. As $\varepsilon_1$ increases, the PSNR performance increases as well. It is because for low $\varepsilon_1$, many correct MVs are identified as unreliable and are merged with improper neighboring MVs for



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

Figure 3.6: According to the different thresholds used in the MV residual classification, (a), (b), (c), and (d) are PSNR performances for Foreman, Fast Food, Formula 1, and Stephan, respectively.

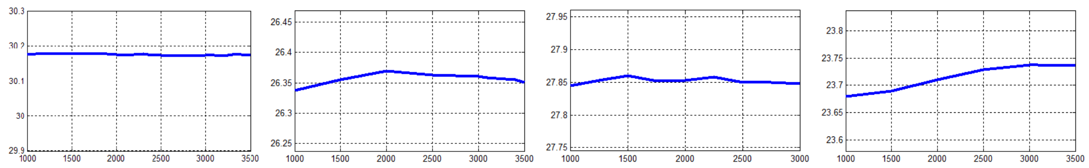
the MV selection. In such cases, the MVF will be rearranged badly that makes it difficult to correct unreliable MVs in the MV refinement and MV smoothing steps. Therefore, the chosen parameter of the residual classification should be low enough to identify as many possibly unreliable MVs as possible, but at the same time, it should be high enough for not merging too many improper MBs as a merged group. Note that we do not show the simulation results of either extremely low or extremely high threshold values in Fig. 3.6. It is because extremely low value case means that the received MVF will be modified thoroughly by MV selection using 32×32 block size and extremely high value case means direct MCFI where the received MVF is used directly. Obviously, both scenarios cannot yield good interpolation results.



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

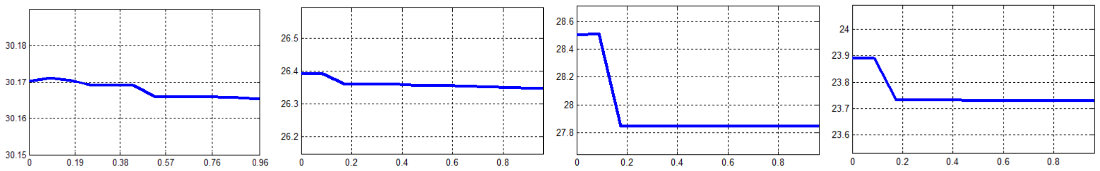
Figure 3.7: According to the different thresholds used in the MV selection, (a), (b), (c), and (d) are PSNR performances for Foreman, Fast Food, Formula 1, and Stephan, respectively.

The simulation results in Fig. 3.7(a)-(d), we observe that the chosen parameter for $\varepsilon_2$ does not affect the PSNR performance considerably. That is, as the resulting MVF becomes static, the MV with minimum ABPD will still be selected even though its ABPD is greater than $\varepsilon_2$. Similarly, if $\varepsilon_2$ is too large, the MVs



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

Figure 3.8: According to the different thresholds used in the BPD energy classification, (a), (b), (c), and (d) are PSNR performances for Foreman, Fast Food, Formula 1, and Stephan, respectively.



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

Figure 3.9: According to the different thresholds used in the MV refinement, (a), (b), (c), and (d) are PSNR performances for Foreman, Fast Food, Formula 1, and Stephan, respectively.

with minimum ABPD are also assigned to all merged blocks. The reason why we place a threshold value for MV selection is to check if there will be new MVs propagating to the neighborhood.

From Fig. 3.8(a) to (b), it also shows that the scale of $\varepsilon_3$ does not have too much influence on MV reclassification since the energy check of MV refinement can avoid choosing unsuitable MVs. However, this threshold value has to effectively detect all significant visible artifacts introduced by MV selection and other possible unreliable MVs that are not identified yet during the residual classification. As to the angle threshold for the MV refinement, $\varepsilon_4$, we observe that the PSNR performance degrades gradually in Fig. 3.9(a)-(d) when the angle difference increases. The reason is that the MV choices for MV refinement become less. We also notice that zero angle difference does not obtain the best average PSNR. This is because similarity check can help us avoid getting same MVs to correct unreliable MVs especially on motion boundaries.

To get the best MCFI performance, these threshold values should be adjustable to different video sequences. However, From the simulation results in Fig. 3.6-3.9, we observe that the proposed MV processing method is only sensitive to $\varepsilon_1$ and $\varepsilon_4$. In order to reduce the complexity, we use the same parameters for all test video clips in the simulation and set $\{\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4\}$ to be 1100, 45, 2000, and 0.15, respectively. These values are not chosen to optimize MCFI results for only one video sequence, but to have the ability to process general video cases. Therefore, we select the values around saturated points for $\varepsilon_1$ and $\varepsilon_4$ and averaged values for $\varepsilon_2$ and $\varepsilon_3$.

## 3.4 Simulations

In this section, we present simulation results to evaluate the performance of the proposed method. We compare our method with VMF, MV smoothing [1], multi-size block matching algorithm [2], and proposed MV selection from the neighboring MVs by minimizing the BPD. Eight video sequences, Foreman, Formula 1, Walk, Bus, Fast Food, Stephan, Rugby, and Football, of CIF frame resolution are used with original frame rate of 30 frame per second (fps). They are all encoded using H.263, where even frames are skipped to generate video bitstreams of 15 fps. The skipped frames are interpolated at the decoder and they are used to evaluate different interpolation schemes. The rate control function is disabled by fixing quantization parameter (QP) values. The averaged bit rates of all test sequences are 395.77Kbps, 474.50Kbps, 430.39Kbps, 509.43Kbps, 499.36Kbps, 503.10Kbps, 340.90Kbps, and 429.32Kbps for Foreman, Formula, Walk, Bus, Fast Food, Stephan, Rugby and Football, respectively.

The visual comparisons are presented in Fig. 3.10 to Fig. 3.15. Fig. 3.10 and Fig. 3.11 show the visual comparisons for Foreman. In Fig. 3.10, blockiness can easily be seen in Fig. 3.10(c), (d), and (e). The blockiness artifacts can be removed by MV smoothing as shown in Fig. 3.10(c). However, ghost artifacts are then generated because incorrect MVs have impacts on the smoothing process such as the areas around the face. Also, the structure of the building and the tower cannot be maintained. In Fig. 3.10(d), the performance of motion estimation is greatly degraded due to image distortion. MV selection can recover the contour of the face and some edge information as shown in Fig. 3.10(e). However, it is more likely to fail when a frame has repeated structures or fairly smooth areas around the MBs that have edge information. This is because minimizing the BPD may choose a MV that points to the smooth area, which will cause a broken edge as shown on the building.

Fig. 3.11 shows another example of FOREMAN. In Fig. 3.11(e), we can clearly see that the hand is deformed and replaced by the sky because the sky is a smoother area around the hand. Moreover, the structure of the yellow tower is destroyed since each MB searches for its own motion without considering object structures. As to the results using the proposed method in Fig. 3.10(f) and Fig. 3.11(f), these artifacts have successfully been eliminated. The ghost effect in Fig. 3.10(c) and Fig. 3.11(c) does not appear in our results since our proposed method corrects those unreliable MVs before smoothing. Comparing to MV selection, our results can better preserve edge information such as the yellow tower by taking object structures into consideration during the MB merging process. Moreover, due to the unidirectional interpolation scheme, we also perform better than the other methods on the frame boundary. It is because those MBs on the frame boundaries have motion moving to the right direction, and part of the content disappears in the subsequent frame. As we use unidirectional MCFI on the frame boundary, the content mismatched effect can be removed.

As one might notice, our PSNR value is lower than that of MV selection in Fig. 3.10, even though our method has better visual quality. This is because PSNR measures signal fidelity to its original frame, instead of perceived visual quality. Any pixel shifting may cause significant PSNR degradation but we may not see the difference. In addition, our proposed method attempts to refine motion by maintaining structure integrity in order to provide better visual experience, instead of recovering the original pixel values. After all, the users do not know the original frames that are skipped at the decoder as long as the motion is smooth and the images have no visible artifacts. Therefore, we adopt an alternative objective measurement, structure similarity (SSIM) index, for quality assessment [49], which has been used in [26]. It examines degradation of structure information and is less sensitive to the pixel shift effect. In this way, the quality numeric analysis can truly reflect structure integrity in interpolated frames. As we can see in Fig. 3.10, although our PSNR is lower than MV selection, our performance is better in terms of SSIM index. In Fig. 3.11, our performance is better in terms of both PSNR and SSIM.

It is noted that SSIM values are derived by averaging all index values within a frame and therefore the difference between our method and MV selection seems smaller. That is, if we divide a frame into 11×9 units of size 32×32 for CIF video sequences, the obtained index value is an average of 99 local index values. Hence the averaging process lowers the influence of local visual artifacts. Even though the difference of SSIM indexes is not much due to the averaging process, our SSIM still performs better than others.

Fig. 3.12 and 3.13 demonstrate two examples in Walk. In Fig. 3.12, because the background has complex patterns, for which it is a challenge to find accurate MVs, the other methods that we compare with fail, and all have severely deformed structures as shown in Fig. 3.12(b), (c), (d), and (e). Our method, however, can recover the background without any artifact even through the frame has many intra-coded MBs and high-residual energy inter-coded MBs. This proves that our MV reliability classification and MB merging can truly help MV processing to obtain a better motion as well as to maintain complex object structures. Our method also outperforms in terms of PSNR and SSIM.

Another interpolation result of a different frame number is demonstrated in Fig. 3.13. As we can see, MV selection can recover many areas except the face where the eyes and nose have been replaced by smooth skin texture. Direct interpolation, vector median filtering, MV smoothing, adaptive VMF, and multi-size block matching algorithm all result in many visual artifacts in both the face and body areas. The proposed method provides much better quality than the other methods by eliminating most of the artifacts as illustrated in Fig. 3.13(f).

The interpolation results of Formula 1 are shown in Fig. 3.14. Fast motion is involved as the camera tries to catch up with the race car and also the

Table 3.2: PSNR performance comparisons among four frame interpolation methods and the proposed multi-stage MV processing method.

| Sequences | Direct | VMF | Smooth | MS BMA | Selection | Multi-Stage |
|---|---|---|---|---|---|---|
| Walk | 22.88 | 22.95 | 22.99 | 23.00 | 23.04 | **23.11** |
| Formula 1 | 28.09 | 28.24 | 27.85 | 28.34 | **28.54** | 28.36 |
| Bus | 22.92 | 23.03 | 22.64 | **24.77** | 23.72 | 24.16 |
| Fast Food | 25.67 | 25.84 | 25.69 | 25.41 | 26.31 | **26.66** |
| Foreman | 31.43 | 31.48 | 31.19 | 31.12 | **31.75** | 30.16 |
| Stephan | 23.80 | 23.87 | 23.68 | 23.77 | **24.09** | 23.93 |
| Football | 24.88 | 24.95 | 24.99 | 23.05 | **25.25** | 25.21 |
| Rugby | 24.58 | 24.75 | 24.76 | 24.45 | **24.97** | 24.81 |

Table 3.3: SSIM performance comparisons among four frame interpolation methods and the proposed multi-stage MV processing method.

| Sequences | Direct | VMF | Smooth | MS BMA | Selection | Multi-Stage |
|---|---|---|---|---|---|---|
| Walk | 0.7869 | 0.7889 | 0.7908 | 0.7911 | 0.7958 | **0.8025** |
| Formula 1 | 0.8276 | 0.8305 | 0.8251 | 0.8363 | 0.8388 | **0.8432** |
| Bus | 0.7893 | 0.7924 | 0.7755 | **0.8690** | 0.8307 | 0.8438 |
| Fast Food | 0.8270 | 0.8304 | 0.8275 | 0.8045 | 0.8566 | **0.8819** |
| Foreman | 0.9311 | 0.9319 | 0.9295 | 0.9237 | 0.9406 | **0.9467** |
| Stephan | 0.7814 | 0.7818 | 0.7743 | 0.7771 | 0.7993 | **0.8116** |
| Football | 0.6752 | 0.6775 | 0.6793 | 0.6634 | 0.7012 | **0.7151** |
| Rugby | 0.7335 | 0.7397 | 0.7416 | 0.7265 | 0.7585 | **0.7694** |

intensity of luminance between grass and pavement is very similar. These factors account for the failed interpolation on the white lines in the background as shown in Fig. 3.14(b), (c), (d), and (e). Bidirectional difference of color components and the MB merging algorithm assist us to find more suitable motion to represent this area so that our white lines are consistent, as illustrated in Fig. 3.14(f).

In order to prove the feasibility of our proposed method, we also apply our algorithm to video sequences with a larger frame size, 720×480, as shown in Fig. 3.15. This sequence is encoded with the bit rate of 1.7Mbps. Obviously, our method still significantly outperforms the other methods.

We list averaged PSNR and SSIM values for these eight video sequences in Table 3.2 and Table 3.3. As observed, our SSIM performance is consistently better than other motion vector processing methods but PSNR performance is only better in Walk, Bus, and Fast Food. PSNR is derived from the quantity of fidelity difference while SSIM emphasizes the similarity of object structures. According to their properties, SSIM is adequate to the MCFI case since avoiding deformed structures and achieving video consistency are the most important requirements in MCFI applications. A counter example is demonstrated in Fig 3.10(f) where our result has better visual quality but PSNR is lower than Fig 3.10(e). However, its SSIM index can more effectively reflect the subjective ratings. For Bus, we perform slightly worse than adaptive VMF and multi-size block matching algorithm. This is because the quality of the re-estimated MVF is much better than the received MVF. Comparing to Direct MCFI, the proposed motion vector processing approach has 1.24dB improvement in average PSNR.

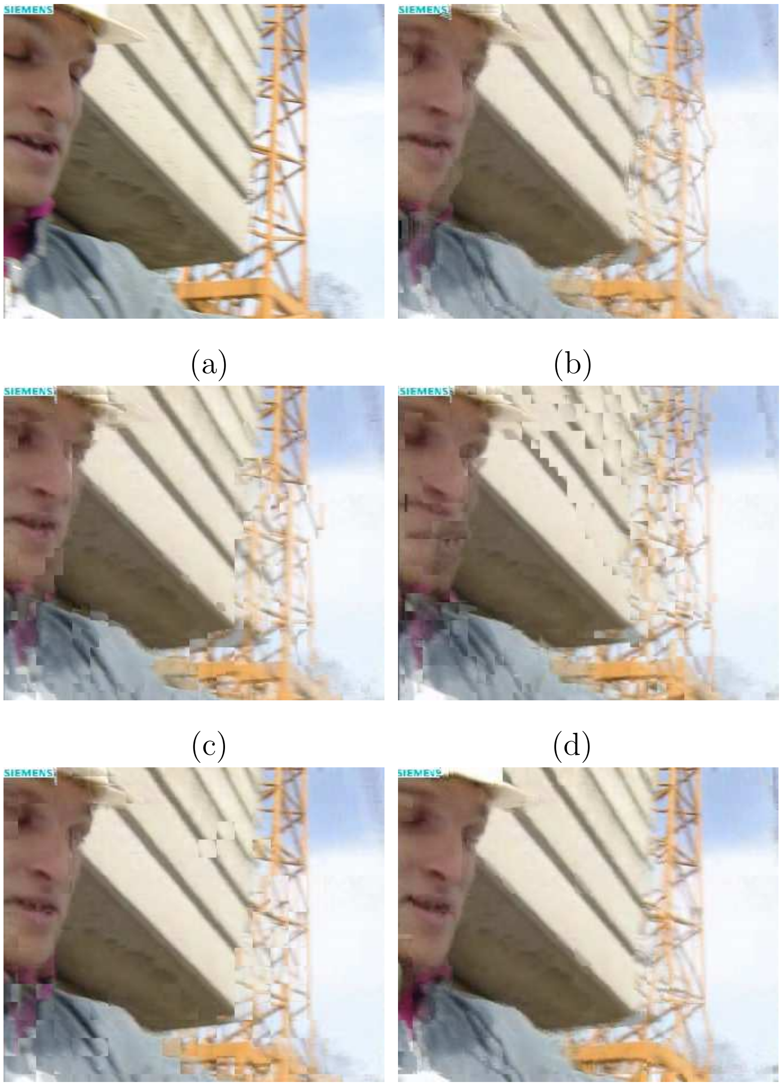
For further comparisons, we plot PSNR and SSIM values of interpolated frames using VMF, MV selection, and the proposed method for the entire Bus sequence, as illustrated in Fig. 3.16. The overall performance of the proposed method is generally the best. Please note that in some difficult frames, where a lot of MVs are unreliable and there are many intra-coded MBs, the proposed method always has less structure degradation. It also seems that our method can keep the performance at a higher level so that the visual quality can be maintained relatively constant during video playback.

Here, we also present simulation results to evaluate the MCFI performance with chrominance processing. Three different methods are used for visual comparisons, direct interpolation, and the proposed method without color consideration and with color consideration. Two video sequences, Foreman and Formula 1, are used for comparison, since their luminance components have much smoother distribution than other clips and general motion estimation methods can easily produce ambiguous motion. The visual comparisons are presented in Fig. 3.17 and Fig. 3.18. In Fig. 3.17(c), although the artifacts around the nose and the eye are reduced using the proposed MV processing method, a lot of unreliable MVs still cannot be identified if we do not consider the color information. These artifacts are removed in the interpolated result as shown in Fig. 3.17(d), since chrominance information sharpens the residual energy and $BPD$ distribution and the unreliable MVs around the shirts and face areas are identified and are corrected accordingly.

In Fig. 3.18, the intensity between grass and pavement is very similar, so unreliable MVs may occur around the white line areas. Comparing to the interpolated results in Fig. 3.18(b) and (c), color components can effectively detect the unreliable MVs and find more suitable motion during the bidirectional MV processing, so the artifacts do not appear in Fig. 3.18(d).

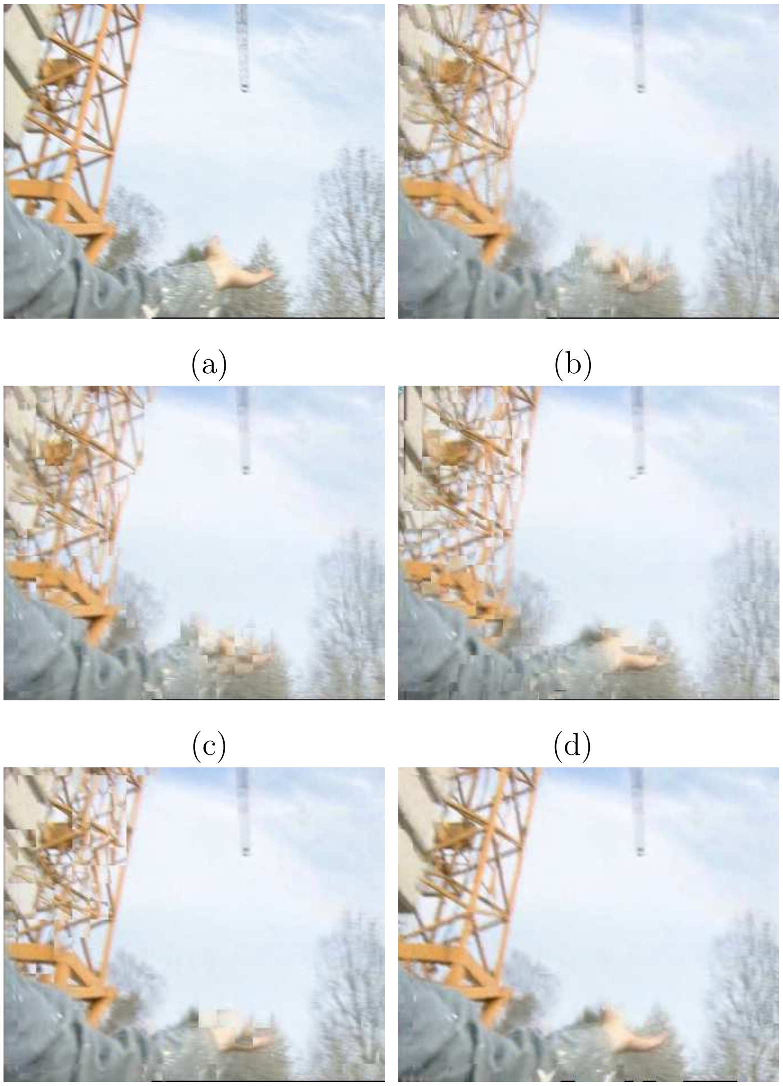
## 3.5 Acknowledgement

Portions of this chapter appear in “A Multistage Motion Vector Processing Method for Motion-Compensated Frame Interpolation,” in *IEEE Transactions on Image Processing*, May 2008, “Motion Vector Processing Based on Residual Energy Information for Motion Compensated Frame Interpolation,” in *IEEE International Conference on Image Processing*, Oct. 2006, “A Novel Motion Compensated Frame Interpolation Based on Block-Merging and Residual Energy” in *IEEE Workshop on Multimedia Signal Processing*, Oct. 2006, “A Novel Multi-Stage Motion Vector Processing Method for Motion Compensated Frame Interpolation,” in *IEEE International Conference on Image Processing*, May 2007, and “Motion Vector Processing Using Bidirectional Frame Difference in Motion Compensated Frame Interpolation,” in *IEEE Workshop on Mobile Video Delivery*, June 2008. The dissertation author was the primary author of these publications, and the listed co-author directed and supervised the research that forms the basis for this chapter.



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

Figure 3.10: The interpolated results of frame 186 of Foreman using (a) original frame, (b) MV smoothing (PSNR: 24.59dB, SSIM: 0.7233), (c) VMF (PSNR: 24.72dB, SSIM: 0.7431), (d) multi-size block matching algorithm (PSNR: 22.44dB, SSIM: 0.5646), (e) proposed MV selection (PSNR: 25.23dB, SSIM: 0.7811), and (f) the proposed multi-stage MV processing method (PSNR: 24.85dB, SSIM: 0.8270).



(a) (b)

(c) (d)

(e) (f)

Figure 3.11: The interpolated results of frame 196 of FOREMAN using (a) original frame, (b) MV smoothing (PSNR: 24.42dB, SSIM: 0.7328), (c) VMF (PSNR: 24.44dB, SSIM: 0.7363), (d) multi-size block matching algorithm (PSNR: 24.76dB, SSIM: 0.7269), (e) proposed MV selection (PSNR: 25.70dB, SSIM: 0.8316), and (f) the proposed multi-stage MV processing method (PSNR: 26.61dB, SSIM: 0.9278).

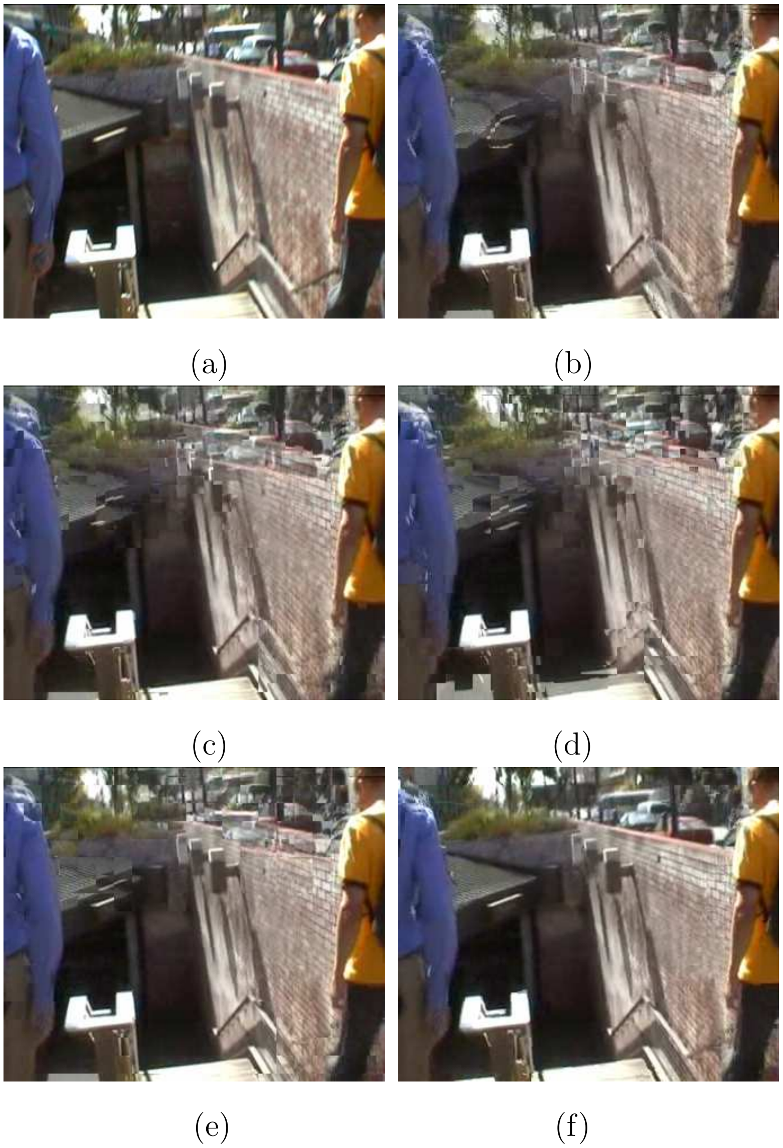
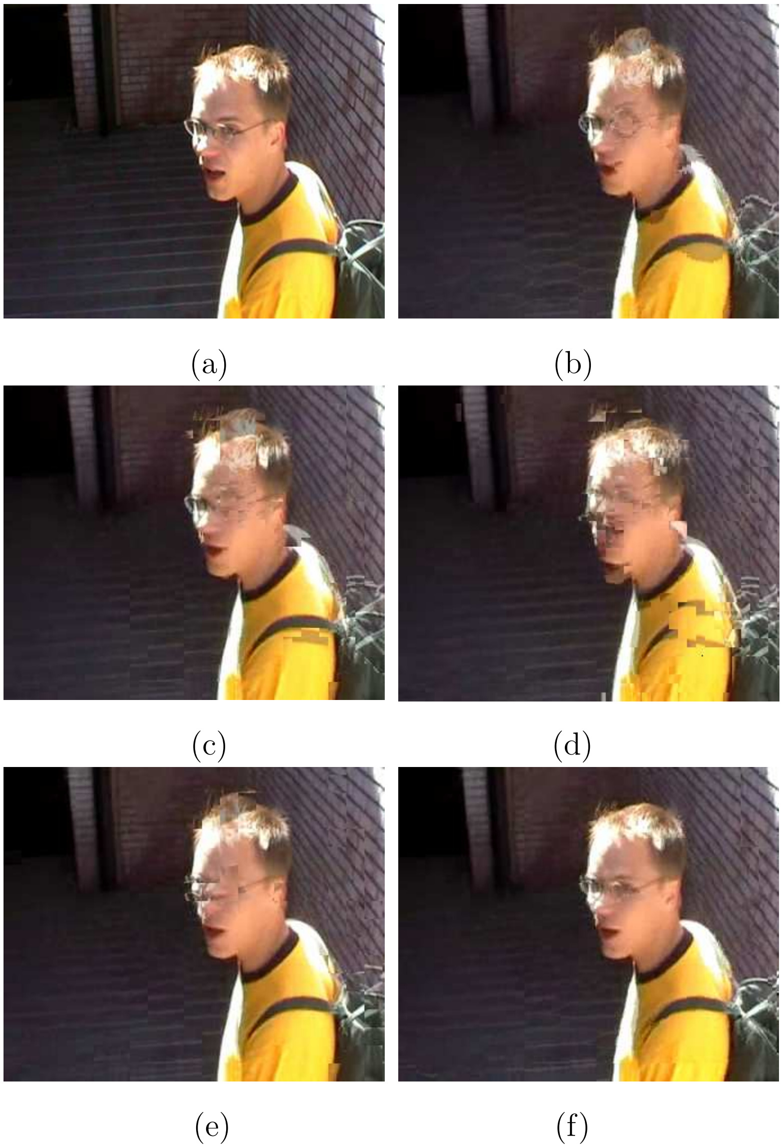


Figure 3.12: The interpolated results of frame 18 of WALK using (a) original frame, (b) MV smoothing (PSNR: 19.76dB, SSIM: 0.6323), (c) VMF (PSNR: 19.67dB, SSIM: 0.6280), (d) multi-size block matching algorithm (PSNR: 19.34dB, SSIM: 0.6234), (e) proposed MV selection (PSNR: 19.94dB, SSIM: 0.6786), and (f) the proposed multi-stage MV processing method (PSNR: 21.33dB, SSIM: 0.8085).

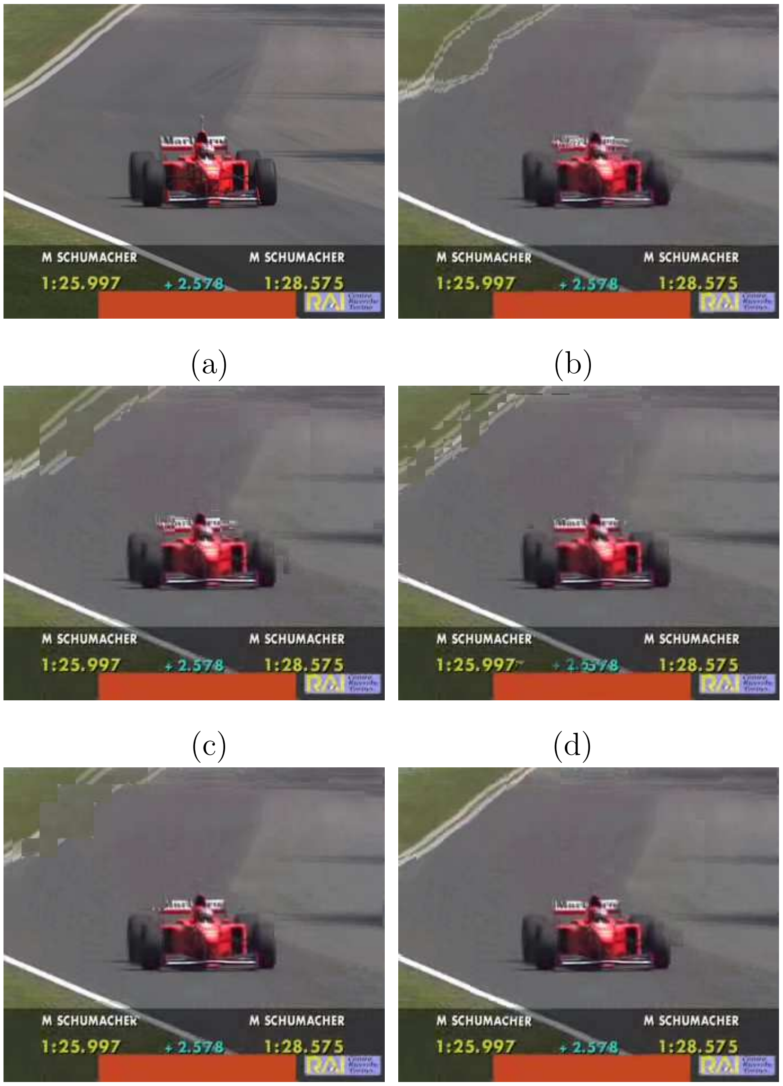


(a) (b)

(c) (d)

(e) (f)

Figure 3.13: The interpolated results of frame 288 of WALK using (a) original frame, (b) MV smoothing (PSNR: 22.35dB, SSIM: 0.7407), (c) VMF (PSNR: 22.04dB, SSIM: 0.7224), (d) multi-size block matching algorithm (PSNR: 21.58dB, SSIM: 0.7024), (e) proposed MV selection (PSNR: 22.22dB, SSIM: 0.7388), and (f) the proposed multi-stage MV processing method (PSNR: 22.39dB, SSIM: 0.7606).

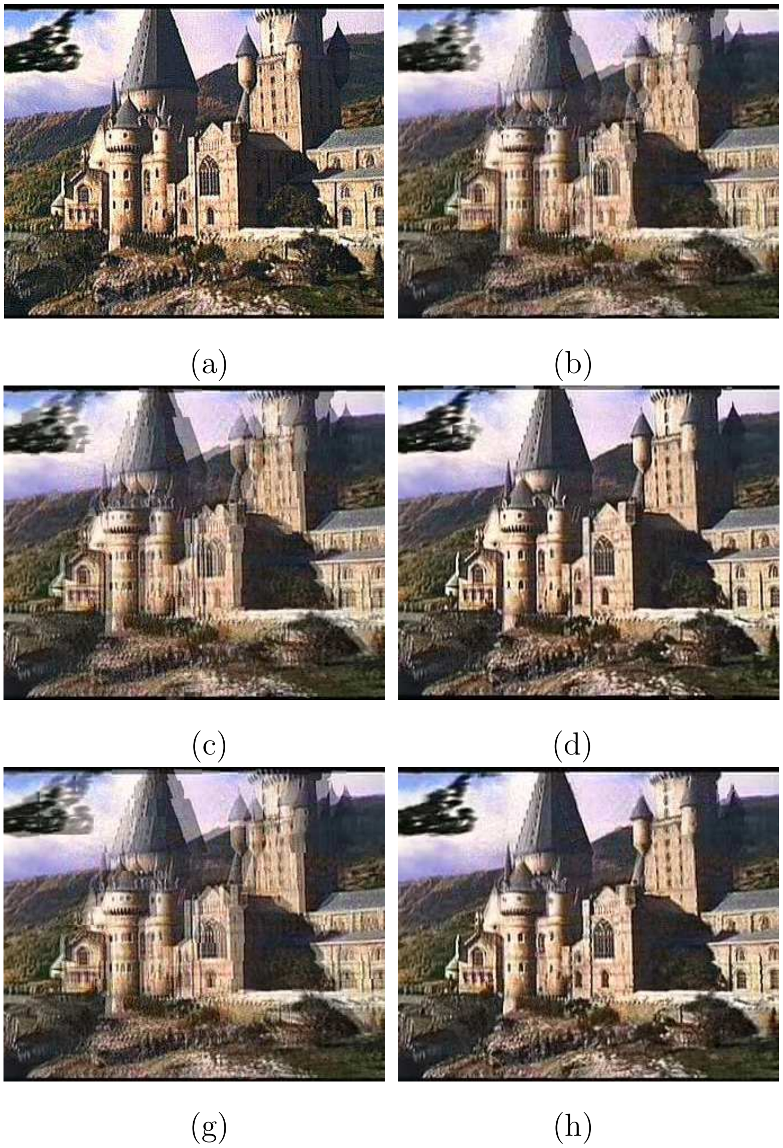


(a) (b)

(c) (d)

(e) (f)

Figure 3.14: The interpolated results of frame 56 of FORMULA 1 using (a) original frame, (b) MV smoothing (PSNR: 29.46dB, SSIM: 0.9013), (c) VMF (PSNR: 29.80dB, SSIM: 0.9067), (d) multi-size block matching algorithm (PSNR: 29.85dB, SSIM: 0.9149), (e) proposed MV selection (PSNR: 30.23dB, SSIM: 0.9225), and (f) the proposed multi-stage MV processing method (PSNR: 31.56dB, SSIM: 0.9567).



(a) (b) (c) (d) (g) (h)

Figure 3.15: The interpolated results of frame 12 of CASTLE AND TREE using (a) original frame, (b) MV smoothing (PSNR: 21.80dB, SSIM: 0.8415), (c) VMF (PSNR: 21.57dB, SSIM: 0.8342), (d) multi-size block matching algorithm (PSNR: 25.62dB, SSIM: 0.9322), (e) proposed MV selection (PSNR: 21.89dB, SSIM: 0.8452), and (f) the proposed multi-stage MV processing method (PSNR: 26.46dB, SSIM: 0.9422).

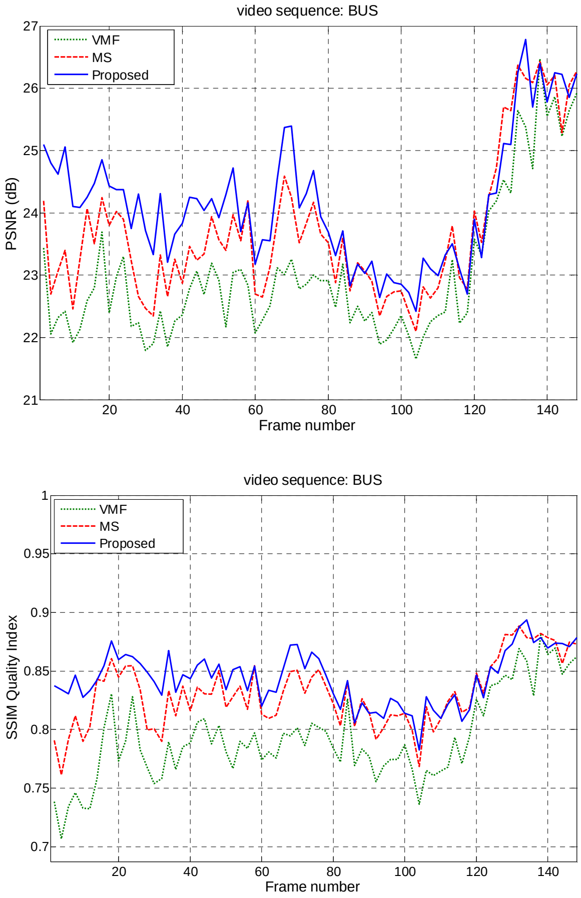
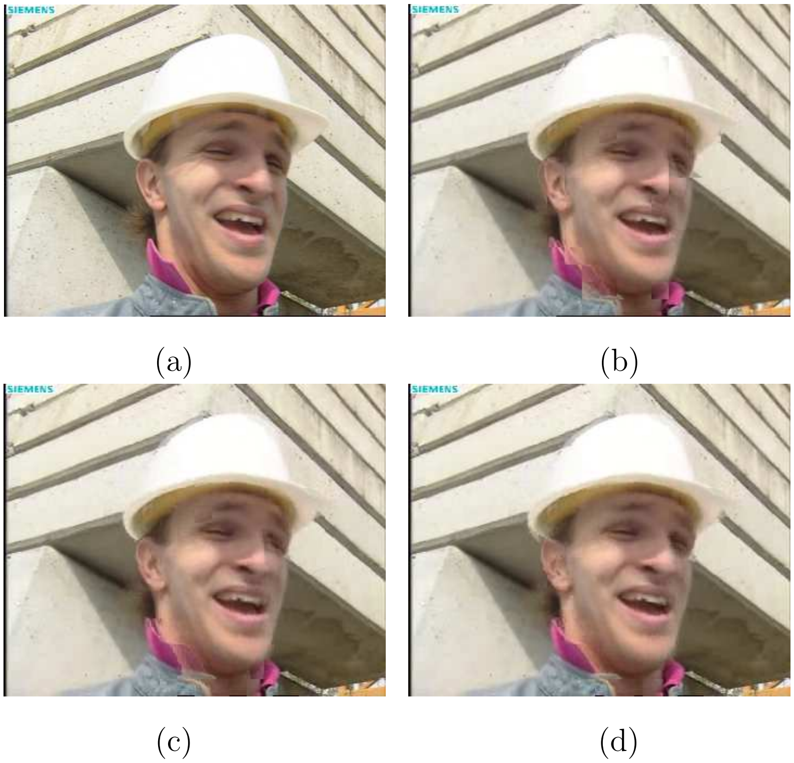


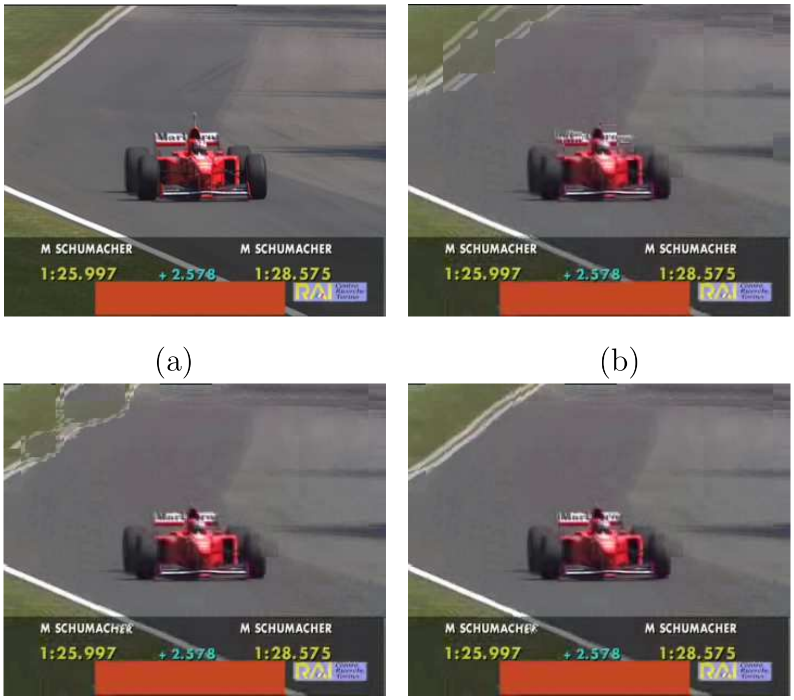
Figure 3.16: The PSNR and SSIM plots for Bus. Green line denotes VMF, red line denotes proposed MV selection with fixed searching size and Blue denotes the proposed multi-stage MV Processing.



(a) (b)

(c) (d)

Figure 3.17: The interpolated results of frame 14 of Foreman using (a) the original frame 14, (b) direct MCFI (PSNR: 32.11dB), (c) the proposed method without color consideration (PSNR: 30.10dB), and (d) the proposed method with color consideration (PSNR: 30.43dB).



(a) (b)

(c) (d)

Figure 3.18: The interpolated results of frame 56 of Formula 1 using (a) the original frame 56, (b) direct MCFI (PSNR: 29.68dB), (c) the proposed method without color consideration (PSNR: 29.58dB), and (d) the proposed method with color consideration (PSNR: 31.56dB).

# 4 Correlation-Based Motion Vector Processing with Adaptive Interpolation Scheme

Although the proposed multi-stage MV processing method in the previous chapter has greatly improved the visual quality for MCFI, we still noticed that not all of the unreliable MVs can be detected by their residual energy and coding types. This usually happens when there are repeated patterns or smooth contents where the encoder may choose MVs that deviate from the real motion but still have very low SAD values. In addition, we adopted the work in [1] as the last MV processing stage to remove blocking artifacts, but it also created ghost artifact on motion boundaries. Finally, we did not explicitly address the problem of video occlusion, which may also cause ghost artifacts.

In this chapter, we further extend the method discussed in Chapter 3 to solve the aforementioned problems. To detect unreliable MVs that have low residual energy, we propose classifying those MVs by calculating MV correlation in their local neighborhood. Unlike VMF that removes MV outliers one at a time and usually fails when irregular MVs occur in a cluster, we merge unreliable MVs due to low MV correlation and select a single best MV from their neighbors. In addition, based on the already calculated MV correlation information, we present an adaptive correlation-based MV smoothing method to remove blockiness artifacts but still keep structure edges. This MV smoothing process can be considered as a weighted averaging process but further takes into account MV correlation to adaptively determine the weights. Moreover, to minimize the ghost artifacts due to video occlusion, an adaptive frame interpolation scheme is proposed to dynamically determine to use forward or backward predictions for occlusion areas based on their surrounding MVs. In these areas, their MVs obtained by minimizing the prediction difference are often unreliable once the occlusion area is larger than a block size used in motion estimation. Therefore, we propose using the neighboring *corrected* MVs outside the occlusion areas but only use either forward or backward prediction. Compared to the methods in [11][17][20], which used image segmentation techniques and traced back several frames to determine occlusions, our method only looks at the current frame and the bidirectional frame difference of each block and hence has lower complexity.

In the following sections, we first demonstrate the MV classification process and the merging process based on the MV correlation for the received MVF in section 4.1. The proposed bidirectional MV processing method is presented in section 4.2, which is different from the previous scheme since it considers the MV correlation and the detection of occlusion areas as well. Finally, the frame interpolation scheme is described in section 4.3.

## 4.1 Motion Vector Correlation Analysis for Motion Compensated Frame interpolation

In addition to the MVs with high residual energies and areas where no MVs are available, the MVs with low correlations can also cause visual artifacts in frame interpolation. Hence, in order to effectively discover all unreliable MVs, the received MVs should be further classified according to different criteria, rather than merely using the residual energy.

Instead of correcting these identified unreliable MVs separately, we propose merging MBs that contain at least one unreliable MV based on the residual distribution or the MV similarity, and then a single best MV can be assigned. Since we use 8×8 block size for the classification process, if an encoder uses 4×4 as the smallest block size for motion estimation such as in H.264 [50], MVs with block sizes smaller than 8×8 will be averaged prior to the classification.

### 4.1.1 Motion Vector Classification

Let $\mathbf{v}_{m,n}$ denote the MV of each 8×8 block. We classify $\mathbf{v}_{m,n}$ into three different reliability levels, *unreliable due to high residual energy*($L_1$), *unreliable due to low inter-MV correlation*($L_2$), and *possibly unreliable*($L_3$). The classification process is similar to the method in Chapter 3, with further analysis of those MVs with low residual energies. First, if the residual energy is greater than a predefined threshold, $\varepsilon_1$, or the coding type is intra-coded, $\mathbf{v}_{m,n}$ will be classified as an unreliable MV and be put into the reliability set $L_1$. For the MVs whose residual energies are less than $\varepsilon_1$, we calculate their MV correlation with neighboring MVs to check if they are unreliable or possibly unreliable. MVs that are very dissimilar to adjacent MVs will be regarded as low correlated MVs and be placed into the reliability set $L_2$. Since the received forward MVs cannot truly represent their co-located MBs in the interpolated frame, all received MVs have to be examined to make sure that they are truly reliable for the bidirectional MCFI scheme. Therefore, for the MVs that are not classified yet, we will put them into the possibly unreliable set $L_3$.

In order to detect irregular MVs, we calculate the correlation index of each MV to all its available adjacent MVs based on 8×8 block size. Here, the correlation index is defined using Euclidian distance between $\mathbf{v}_{m,n}$ and its adjacent MVs. According to our observation, motion magnitude distances usually become higher than other areas if the local movement is relatively large. Therefore, to reduce the sensitivity from motion magnitude values, the correlation index is defined as the magnitude variance in the local neighborhood:

$$C_{m,n} = \frac{\frac{1}{8}\sum_{i=-1}^{1}\sum_{j=-1}^{1}\|\mathbf{v}_{m,n} - \mathbf{v}_{m+i,n+j}\|_2}{\frac{1}{9}\sum_{i=-1}^{1}\sum_{j=-1}^{1}\|\mathbf{v}_{m+i,n+j}\|_2} \tag{4.1}$$

where $\mathbf{v}_{m,n}$ is the center MV and $\mathbf{v}_{m+i,n+j}$ are the surrounding available MVs, i.e. inter-MVs. As observed, if the local magnitude is large, the magnitude distance should be large enough so that the increase of $C_{m,n}$ is noticeable. For the remaining MVs with low residual energy, to determine if they are unreliable or possibly unreliable, we compare $C_{m,n}$ with the averaged MV correlation index in this neighborhood, which can be written in the following:

$$C_{m,n}^{avg} = \frac{1}{9} \sum_{i=-1}^{1} \sum_{j=-1}^{1} C_{m+i,n+j} \tag{4.2}$$

If $C_{m,n}$ is greater than $C_{m,n}^{avg}$ and the motion distance is greater than half of the averaged magnitude, $\mathbf{v}_{m,n}$ will be considered as an unreliable MV. For the MVs that are not classified yet, we place them in the reliability set $L_3$. The MV reliability map can therefore be created as follows:

$$MVRM_1(m,n) = \begin{cases} L_1, & \text{if } E_{m,n} \geq \varepsilon_1, \\ L_2, & \text{if } C_{m,n} > C_{m,n}^{avg} \text{ and} \\ & C_{m,n} > 0.5, \\ L_3, & \text{otherwise.} \end{cases} \tag{4.3}$$

Here, since the unreliable MVs are classified due to different criteria, we place unreliable MVs caused by high residual and low MV correlation into different reliability sets, $L_1$ and $L_2$, respectively. Based on this MV reliability analysis, the MV residual merged map and the MV correlation merged map can then be produced to assist the following MV correction processes.

### 4.1.2 Macroblock Merging Map for Motion Vector Processing

Previously, we suggested that unreliable MVs should be grouped into larger blocks for MV correction according to the residual energy distribution and predefined merging shapes. We also apply this algorithm to all MVs in $L_1$ to create a MV residual merging map. However, since MVs of $L_1$ and $L_2$ are identified due to different reasons, they should not be merged together. Most conventional methods prefer using VMF to correct the MVs in $L_2$, but this method can easily fail if low correlated MVs occur in clusters. If we assume that adjacent highly correlated MVs should belong to the same object, then we should not correct these irregular MVs separately, but merge those MBs having similar irregular MVs as a merged group for further MV correction. From the observation of the received MVF, irregular MVs often come from scenes composed of repeated or texture-like patterns, or smooth contents. As such, the MV selection process, which will be described in the next section, is more likely to choose the right motion using a merged group. This is because a larger block size has more pixel references than smaller ones so that the estimation results are less likely to be affected by scene content.

Based on the proposed MV analysis, we can create two reference merging maps so that adjacent unreliable MVs can be corrected together and yield a single best motion. The merging process is performed on a MB basis, and all MBs that contain unreliable MVs will be examined in a raster scan order. The residual MB merging map, $MBMM_r$, is created based on residual energy distribution. A separate merging map, $MBMM_c$, is created for unreliable MVs in $L_2$. If we find a MB containing unreliable MVs in $L_2$, adjacent MBs that have not yet been merged will be checked to see whether they have MVs similar to these unreliable MVs. If this is the case, these MBs will be merged together. If there are no similar MVs in the neighborhood, this MB will remain as a single 16×16 block and this unreliable MV is regarded as an isolated MV. MVs are considered as similar if their angular distance, $d_\theta$, and Euclidian distance are less than predefined thresholds, $\varepsilon_\theta$, and $\varepsilon_m$, respectively. Here, the angular distances $d_\theta$ can be represented as:

$$d_\theta = 1 - \frac{\mathbf{v}_{m,n} \cdot \mathbf{v}_{i,j}}{|\mathbf{v}_{m,n}||\mathbf{v}_{i,j}|} = 1 - \cos_\theta, \tag{4.4}$$

where $\theta$ is the angle difference between $\mathbf{v}_{i,j}$ and $\mathbf{v}_{m,n}$.

We choose the same maximum block size as in Section 3.1 for the merging process. In both merging maps, each merged group will be assigned a unique index number to the MBs belonging to the same group. For the MV reliability classification process, the best value of $\varepsilon_1$ was found empirically in Section 3.3 as 1100 and we also use the same value in the simulation.

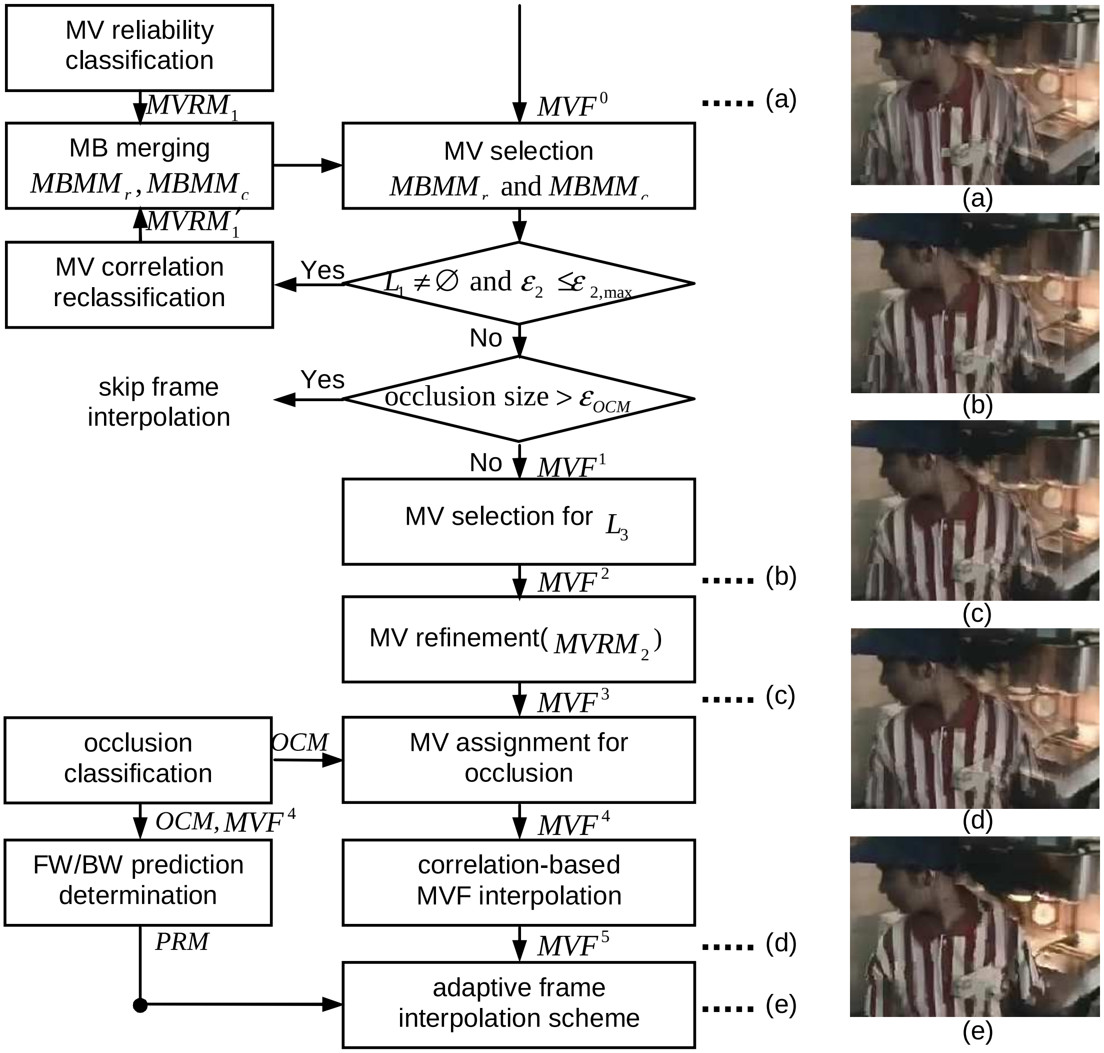

MV reliability classification
$MVRM_1$
MB merging $MBMM_r, MBMM_c$
$MVRM_1'$
MV correlation reclassification
Yes
skip frame interpolation
Yes
$MVF^0$
..... (a)
MV selection $MBMM_r$ and $MBMM_c$
$L_1 \neq \emptyset$ and $\varepsilon_2 \leq \varepsilon_{2,\max}$
No
occlusion size $> \varepsilon_{OCM}$
No
$MVF^1$
MV selection for $L_3$
$MVF^2$
..... (b)
MV refinement($MVRM_2$)
$MVF^3$
..... (c)
occlusion classification
$OCM$
MV assignment for occlusion
$OCM, MVF^4$
$MVF^4$
FW/BW prediction determination
correlation-based MVF interpolation
$PRM$
$MVF^5$
..... (d)
adaptive frame interpolation scheme
..... (e)
(a)
(b)
(c)
(d)
(e)

Figure 4.1: Block diagram of the proposed algorithm. $MVF^k$ and $MVRM_1'$ are the updated motion vector field and the updated MV reliability map, respectively. (a), (b), (c), (d), and (e) are the interpolation results using the original received MVF, the MVF after MV selection, the MVF after the MV refinement, the MVF after the MV averaging, and the adaptive MCFI scheme, respectively.

## 4.2 Correlation-Based Motion Vector Processing using Bidirectional Prediction Difference

According to the residual merging map ($MBMM_r$), the correlation merging map ($MBMM_c$), and the received MVF, we can select the best MV for each merged group from its own and neighboring MVs by minimizing the $ABPD$. As shown in Fig. 4.1, this proposed MV selection process is performed along with an iterative threshold mechanism to decide when the process should be terminated. Initially, only the MV whose $ABPD$ is less than the predefined threshold value, $\varepsilon_2$, will be selected to correct unreliable MVs within each merged group. If this is not the case, these unreliable MVs will remain the same and wait for the future correction. To trace the MV correction status, during each MV selection pass, only the index numbers of the corrected MVs and their associated merged MBs will be cleared from the merging maps and the MV reliability map. If there are still non-zero indices in the updated $MBMM_r$ and the MVF status is no longer changed due to a limitation on the threshold value, $\varepsilon_2$ will be increased for the next MV correction pass. Since a image may consist of various contents such as objects with constant intensity (i.e. low $ABPD$ value) and sharp edges (i.e. high $ABPD$ values), by adaptively adjusting threshold values, we can gradually choose the best motion for each merged group. In this way, not only can better motion with lower $ABPD$ value propagate to the neighborhood of unreliable MVs during MV selection, but unreliable MVs can be corrected according to their degree of MV reliability, i.e. $ABPD$ values.

For the subsequent MV selection pass, since the correlation distribution has been changed, the irregular MVs that are previously classified in $L_3$ can be detected. This is because low-correlated MVs usually appear in a cluster and the initially detected unreliable MVs are probably located on the boundary where irregular MVs start to occur. As shown in Fig 4.1, to correct these unreliable MVs, we recursively examine the updated MV correlation distribution and update the correlation merging map, $MBMM_c$, accordingly. The subsequent MV correlation classification will skip the unreliable MVs of $L_1$ that are not corrected in the previous pass. This is because their merging status is determined by residual energy distribution rather than motion correlation. That is, for the unreliable MVs in $L_1$, if their reliability level is not changed, their merging status will remain the same.

The MV selection process stops whenever the merged groups in $MBMM_r$ are all assigned a single best motion, or $\varepsilon_2$ is greater than a predefined maximum threshold value, $\varepsilon_{2,max}$. This maximum threshold value should be designed to find appropriate motion for all merged groups. So, if there are merged groups of $MBMM_r$ that are not assigned any motion due to high $ABPD$ values, they can be considered as occlusions. We will leave their MVF unchanged for now, since forcing them to have new MVs using a very high $\varepsilon_2$ value still cannot obtain reliable MVs. For these occlusion areas, a parallel occlusion classification is undertaken for further processing. The reason why the iteration process is only defined based on $MBMM_r$ is that most occlusions occur in high residual energy and intra-coded areas. Here, we simply use a fixed $\varepsilon_2$ value at 45 for unreliable MVs in $L_2$, which was obtained from experimental simulations in Section 3.3. For unreliable MVs in $L_1$, we start the $\varepsilon_2$ value from 30 with the step size of 15, and set the $\varepsilon_{2,max}$ value to be 60 for all test sequences.

As shown in Fig 4.1, to ensure the remaining unchecked MVs, i.e. possibly unreliable MVs, are truly reliable, we apply the MV selection process with fixed block size of 16 ×16 to examine if their MVs do have smaller $ABPD$ than others. Since the MVF is corrected and regular at this stage, even with a small fixed block size, the possibility to select inaccurate MVs for the MVs in $L_3$ is relatively low. As MV selection always prefers the major motion for each merged group, once selected motion cannot well represent detailed areas such as motion boundaries, areas with different motion usually have higher $ABPD$ values than other areas. Therefore, we classify the new obtained MVF based on the smaller 8×8 block size using $ABPD$ energy distribution. After the new identified unreliable MVs in $MVRM_2$ are further refined using the MV refinement method in Section 3.2, to minimize the blockiness artifacts and also keep the object edge sharp at the same time, we resample the 8×8 MVF into finer 4×4 MVF with consideration of both

MV correlation and $ABPD$ distribution. The following subsections will present in detail the MV selection process and the MVF interpolation process. In Fig 4.1, we also demonstrate how the received MVF is gradually corrected and improved based on the interpolated results for each stage.

### 4.2.1 Motion Vector Selection

We take $MBMM_r$ and $MBMM_c$ as initial reference maps for the MV correction process. In Eqn. (3.3), the MV selection process selects the best MV, $\mathbf{v}_b^*$, for each merged group based on the minimum averaged absolute difference between forward and backward predictions. For the merged groups indicated by $MBMM_r$, we simply select the best motion using Eqn. (3.3). However, those unreliable MVs in $L_2$ are identified due to irregular MV distribution. Depending on what the scene is composed of and how MV estimation is performed at encoder, the selected MVs may tend to distribute randomly if we merely consider the minimum $ABPD$. Hence, we take both minimum $ABPD$ and MV correlation into account for merged groups of $MBMM_c$. That is, we choose MVs that have minimum $ABPD$ among several MV candidates. Adjacent MVs that have higher correlations than the original MV will be chosen as a candidate, and therefore the MV candidate set, $S$, can be re-written as follows:

$$S = \begin{cases} \mathrm{v}_{i,j}, & \text{if } C(\mathrm{v}_{i,j}) < C(\mathrm{v}_{m,n}) \\ \emptyset, & \text{otherwise.} \end{cases}$$

Slightly different from Eqn. (4.1), the correlation index, C(v), is calculated based on the boundary MVs of the merged group and its neighboring MVs. That is, each merged group is considered as a unit block, and only the motion distances between the merged group and its neighboring available MVs are used to select MV candidates. This is because the MVs are similar or equal within the merged group so that their motion distances are too small to be considered. Directly calculating the motion distance may not truly reflect the local motion correlation for each merged group. If the correlation index of $\mathrm{v}_{i,j}$ is less than the original correlation index, $\mathrm{v}_{m,n}$, $\mathrm{v}_{i,j}$ will then be considered as MV candidates for $S$. Once the best

MV exists, we assign it to all MBs within the merged group. If not, it means that there are no other MVs that have higher correlation and can better represent the local motion than the original MVs. In such a case, these unreliable MVs might belong to an area where the motion starts to differ or has different moving objects. We therefore skip this MV selection process and keep the MVs and the $MBMM_c$ unchanged.

Please note that the MV selection processes for both types of unreliable MVs are performed in the same pass. Hence, if the MV assignment for $MBMM_r$ is not completed due to the threshold mechanism, we will have to update $MBMM_c$ according to the current MV correlation distribution for the next pass. That is, as the unreliable MVs have been corrected and their corresponding index numbers have been cleared from $MVRM$, $MBMM_r$, and $MBMM_c$, MV correlation is re-analyzed and an updated MV map can be created as follows:

$$MVRM_1'(m,n) = \begin{cases} L_1, & \text{if } MVRM_1(m,n) = L_1, \\ L_2, & \text{if } C'_{m,n} > C^{avg'}_{m,n} \text{ and} \\ & C'_{m,n} > 0.5, \\ L_3, & \text{otherwise.} \end{cases} \tag{4.5}$$

If unreliable MVs of $L_1$ are not corrected yet, these MVs with high residual energies will still be put into $L_1$ level for the next correction. $C'_{m,n}$ and $C^{avg'}_{m,n}$ represent updated correlation index and averaged correlation index for the latest correlation distribution, respectively. According to this motion re-analysis, we can discover more low-correlated unreliable MVs that have not been detected in the first place. Once $MVRM_1'$ is updated, the corresponding $MBMM_c$ can be recreated as well. However, the merging status of $MBMM_r$ will be the same except for the blocks whose MVs have been corrected. At the end of each pass, we will check if all unreliable MVs of $L_1$ are corrected, if the threshold value is still within the predefined range, and if the occlusion caused by unreliable MVs of $L_1$ has reasonable size, to decide when the MV selection process should be completed. In Fig 4.1, due to the consideration on MV correlation, the unreliable motion in the direct interpolated result has been corrected. The ghost artifacts around the shirt area do not appear in the interpolated result using $MVF^2$.

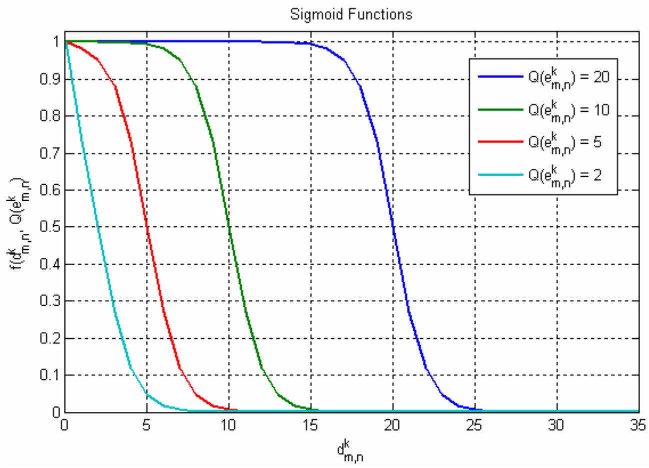


Figure 4.2: The sigmoid functions based on different $ABPD$ values for the MV smoothing. x axis represents MV magnitude distance and y axis represents corresponding weights for MV averaging.

Although the proposed MV selection process needs to recursively calculate $ABPD$ for each merged group, MV candidates are almost the same except for the new MVs that just propagate to the neighborhood. In order to reduce the computational complexity, we actually create two tables to save MVs that have occurred before and their corresponding $ABPD$ values to avoid the repeated calculation. In this way, the required computational complexity is similar to the previously proposed method in Chapter 3.

### 4.2.2 Adaptive Motion Vector Averaging Based on MV Correlation

To reduce blockiness artifacts, we further resample the MVF from one MV with 8x8 block size, $\mathbf{v}_{m,n}$, into four MVs, $\{\mathbf{v}^1_{m,n}, \mathbf{v}^2_{m,n}, \mathbf{v}^3_{m,n}, \mathbf{v}^4_{m,n}\}$. In general, a vector averaging filter can always provide desirable MV smoothing effect for reducing blockiness artifacts. However, the visual quality of the motion sensitive areas such as sharp object edges and striped textures is often distorted by the MV smoothing. This is because in these areas, unpleasant artifacts can easily show up even when motion is only modified slightly. Therefore, motion smoothing should be performed by taking motion correlation and scene contents into consideration so that we can reduce smoothing impacts from neighboring MVs in these motion sensitive areas. Based on this argument, we propose an adaptively weighted vector averaging process as follows:

$$\mathbf{v}^k_{m,n} = \left[ \frac{\sum\limits_{i,j} f(d^k_{m,n}, e^k_{m,n}) v_x}{\sum\limits_{i,j} f(d^k_{m,n}, e^k_{m,n})}, \frac{\sum\limits_{i,j} f(d^k_{m,n}, e^k_{m,n}) v_y}{\sum\limits_{i,j} f(d^k_{m,n}, e^k_{m,n})} \right], \tag{4.6}$$

where $v_x$ and $v_y$ are horizontal and vertical components of $\mathbf{v}_{i,j}$, respectively. Here, $\mathbf{v}_{m,n}$ and bidirectional prediction difference energy, $BPD_{m,n}$, are partitioned into four sub-blocks, $\mathbf{v}^k_{m,n}$ and $e^k_{m,n}$, individually. Please note that $BPD_{m,n}$ is obtained during the MV reclassification process and the MV refinement process. $e^k_{m,n}$, which is the same as $BPD_{m,n}$, is used to roughly measure the interpolation difficulty. That is, if a moving object is not exactly the same in two consecutive decoded frames, areas where object is distorted should have high $BPD_{m,n}$ values. As such, motion smoothing can help to minimize the difference between block boundaries. When scenes consist of simple textures or scenes between two decoded frames are the same, the weights of adjacent MVs should be decreased since the scene content might be very sensitive to MV adjustment. $d^k_{m,n}$ is the corresponding Euclidian distance between $\mathbf{v}^k_{m,n}$ and adjacent MVs, $\mathbf{v}_{i,j}$. If the distance is large, which usually happens when motion has sudden change, the corresponding weights should be reduced to reserve sharp object edges. Based on this discussion, we therefore choose $f$ function as an inverse mapping function for both vector distance and $BPD_{m,n}$.

Initially, we assign $\mathbf{v}_{m,n}$ to $\mathbf{v}^k_{m,n}, k = 1, 2, 3, 4$. Then, we set the weight for the centered MV to be one and the weights of neighboring $\mathbf{v}^k_{m,n}$ will be updated individually using Eqn. (4.6). We choose sigmoidally shaped function for Eqn. (4.6) with two input parameters, $d^k_{m,n}$ and $e^k_{m,n}$, to adaptively adjust the weights for MV averaging. Hence, the inverse mapping function can be then written as follows:

$$f(d^k_{m,n}, Q(e^k_{m,n})) = \frac{1}{1 + e^{(d^k_{m,n} - Q(e^k_{m,n}))}} \tag{4.7}$$

where $Q(e^k_{m,n})$ is the step function of $BPD_{m,n}$. In Eqn. (4.7), the sigmoidal function opens to the right, so as $d^k_{m,n}$ increases, the weight value decreases accordingly. $Q(e^k_{m,n})$ is used to decide the center of the sigmoidal function in which the weight value reduces to half. Based on the previous discussion, we should shift the sigmoidal center rightward when $BPD_{m,n}$ becomes large. Similarly, as $BPD_{m,n}$ decreases, the center will be moved leftward until $BPD_{m,n} = 0$ and only same MVs can have non-zero weights, i.e. 1. In Fig. 4.2, we use four different sigmoidal functions for MV averaging in the experimental simulations. As observed, MVs whose distances are similar within a certain range can have same or similar impacts during vector averaging. Likewise, once motion magnitude distance goes beyond a certain range, we can also reduce its weight immediately.

In our implementation, we actually sample the sigmoidal functions with four different $Q(e^k_{m,n})$ values and save these sampled values in look-up tables. Hence, without calculating the actual exponential function, the proposed MVF interpolation method can simply obtain corresponding weights from the table according to $d^k_{m,n}$ values. The performance gain for the adaptively weighted MV averaging is shown in Fig 4.1. As observed, the blockiness artifacts around areas where $ABPD$ energy is high are removed and the object contour such as the face also looks sharp.

## 4.3 Adaptive Frame Interpolation Scheme for Occlusion Areas

As mentioned in the previous section, we do not assign any MVs to occlusion areas since the motion is only reliable when appropriate predictions can be found from both forward and backward frames. Hence, if the unreliable MVs in $MBMM_r$ have not been corrected until the MV selection process terminates, the MBs that still have non-zero indices will be regarded as occlusions. In addition, the MBs whose MVs still cannot be corrected during the MV refinement stage are also considered as occlusions. In order to assist the subsequent frame interpolation in occlusion areas, an occlusion map $(OCM)$ is created to indicate the occlusion position and range as follows:

$$OCM_{m,n} = \begin{cases} 1, & \text{if } MBMM_r(m,n) \neq 0, \\ 2, & \text{MB whose } \mathrm{BPD_{m,n}} > \varepsilon_3, \\ 0, & \text{otherwise} \end{cases} \tag{4.8}$$

where $\varepsilon_3$ is the same as the threshold value in Section 3.3. In $OCM$, the first type of occlusion often has larger size and the resulting artifacts are more visible, such as deformed structures and the occurrence of the new objects. The second type of occlusion is often the motion boundary or the surrounding MBs of type 1 occlusions. As shown in Fig. 4.1, we calculate the occlusion size to determine whether we are capable of recovering occlusion regions. If the occlusion size is larger than a predefined threshold, $\varepsilon_{OCM}$, we will skip the interpolation process and repeat the current decoded frame.

Excluding the appearance of new large objects and large-scale object distortion, most occlusion cases are commonly caused by existing moving objects. These occlusions usually occur around object contours or frame boundaries, so their sizes are often within a reasonable range and we can easily recover them by adaptively selecting the forward or backward prediction. By analyzing the motion distribution around occlusion areas, a prediction reference map can be further created to determine whether forward prediction or backward prediction is better for occlusion areas.

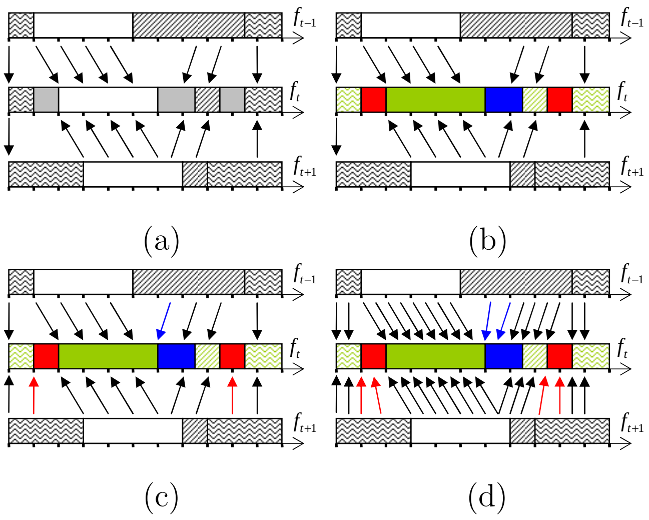
### 4.3.1 Adaptive Frame Interpolation Scheme

To reduce the possible visual artifacts in the bidirectional interpolation scheme, Eqn. (2.2) is modified as follows:

$$f_t(i,j) = w_f \cdot f_{t-1}(i + \frac{1}{2}v_x, j + \frac{1}{2}v_y) + w_b \cdot f_{t+1}(i - \frac{1}{2}v_x, j - \frac{1}{2}v_y)$$

where, $w_f$ and $w_b$ are the weights for forward and backward predictions, respectively. In the identified occlusion areas, $w_f$ and $w_b$ should be adaptively adjusted to obtain the best visual experience.

By observing the corrected MVF, we can learn about the motion distribution and further analyze how these movements cause the occlusion. We again use



(a) (b)

(c) (d)

Figure 4.3: The occlusion process for general video contents. (a) the MVF for non-occlusion areas where occlusion areas are denoted by gray color (b) the prediction reference map ($PRM$) (c) the new obtained MVF after assigning neighboring correct MVs to occlusion areas, and (d) the finer MVF after the MV smoothing stage.

Fig. 2.3 as an example. As observed, the upper occlusion (uncovered region) is induced by $f_{t-1}$, so the better prediction can only be obtained from the backward frame, $f_{t+1}$. Similarly, the lower occlusion (covered region) can only find correct prediction in the forward frame, $f_{t-1}$. That is, the co-located blocks of the moving object's original position should have backward predictions, and the co-located blocks of the moving object's current position should have forward predictions. Hence, the indications of the prediction selection can be differentiated by examining MV directions and MV magnitudes, and a block-based prediction reference map ($PRM$) can be derived:

$$PRM_{m,n} = \begin{cases} w_f = \frac{1}{2},\ w_b = \frac{1}{2}, & \text{if } OCM_{m,n} = 0, \\ w_f = 1,\ w_b = 0, & \text{if } b_{m,n} \text{ of } f_{t-1} \text{ is} \\ & \text{pointed by MVs} \\ w_f = 0,\ w_b = 1, & \text{otherwise.} \end{cases} \tag{4.9}$$

In our implementation, we examine the neighboring MVs for each occlusion area in forward direction. If co-located blocks of the occlusion region are not pointed by any correct MVs, this occlusion region will only have backward predictions for frame interpolation. Otherwise, we assign forward predictions. Same criteria can also be used to explain the occlusion occurrence on frame boundaries. If a frame boundary MB does not have any MVs pointing to it, obviously, it is one of the initial blocks of the whole movement. In such case, the backward prediction should be considered.

In Fig. 4.3, we use a synthesized figure to further demonstrate this occlusion process. In Fig. 4.3(a), there are three moving objects, which are indicated using different structure patterns, and each grid represents 8×8 block size. As observed, the received MVF has been corrected for non-occlusion areas, but for the occlusion areas, which is indicated by gray color, there is no motion since it is very difficult to find predictions bidirectionally. By analyzing the motion distribution on non-occlusion areas, the resulting prediction map is shown in Fig. 4.3(b) where red color denotes using backward predictions, blue color denotes using forward predictions, and green color denotes using bidirectional predictions. Since the proposed interpolation scheme uses three different prediction modes for frame interpolation, to make the pixel values do not change abruptly on boundaries between forward and backward predictions, we apply low-pass filtering on the boundary pixels afterward.

### 4.3.2 Motion Vector Processing in Occlusion Areas

As shown in Fig. 4.1, the type 1 occlusion regions are not assigned motion until the object major motion and detailed motions are determined. Different from some of the conventional assumptions that only assume one motion (background is static) or two motions around occlusion areas, occlusion usually has various movements. As a result, it is very difficult to tell which movements a occlusion should have once its size is large. In order to avoid occlusion motion distribution from being dominated by either motion, we spirally assign MVs to occlusion regions. Along the spiral trace, we start from the left-top and clockwise assign MV to each occlusion block using VMF based 8×8 block size. If there are no MVs available for the VMF process, we will check this block in the future iteration. The assignment stops until all occlusion blocks have MVs. In this way, once the occlusion size is large, we can ensure these interpolated blocks can still follow the neighboring movement.

We use Fig. 4.3(c) for the further demonstration, in which the occlusion blocks have been assigned MVs unidirectionally from the neighboring corrected MVs. After the MVF has been further smoothed and interpolated using the proposed adaptively weighted MV averaging filter, the final corresponding MVF and the prediction map will be used for the frame interpolation as shown in Fig. 4.3(d). Please note that this MV processing does not include the type 2 occlusions since they already have motions. The reason why these MVs still have high $BPD$ energies after MV refinement is that they may be located on the edges of the occlusion area. In Fig 4.1, the interpolated result using the proposed adaptive interpolation scheme looks much better than the general bidirectional scheme. The occlusion artifacts around the sleeves, the shoulder, and the name tag are removed.

## 4.4 Simulations

In this section, simulation results are demonstrated to evaluate the performance of the proposed method. We compare our method with direct interpolation, VMF, bidirectional BMA with consideration on MV correlation based on 8×8 and 16×16 block size, and the proposed MV processing method with bidirectional prediction (Bi-Pred) and adaptively selective prediction (A-Pred). For the bidirectional BMA, we set the motion search range to be -16 to 16 and the obtained MV can be written as:

$$\mathrm{v}_{m,n} = \arg\min_{\mathrm{v}=(v_x,v_y)}(SAD(\mathrm{v}) + \lambda \sum_{\mathrm{v}_N} |\mathrm{v} - \mathrm{v}_N|)$$

where $SAD(\mathrm{v})$ is the sum of absolute difference of forward and backward predictions using v. $\mathrm{v}_N$ are the neighboring casual MVs of $b_{m,n}$. We set $\lambda$ to be 40, which is the multiplier for the influence of MV difference. Six video sequences, Foreman, Formula 1, Walk, Fast Food, Stephan, and Football of CIF frame resolution are used with the original frame rate of 30 frames per second (fps). They

Table 4.1: PSNR performance comparisons among five frame interpolation methods and the proposed correlation-based MV processing method.

| Sequences | VMF | Selection | Bi-BMA8 | Bi-BMA16 | Bi-Pred | A-Pred |
|---|---|---|---|---|---|---|
| Walk | 22.95 | 23.08 | 22.07 | 22.17 | **23.14** | 22.97 |
| Formula 1 | 28.75 | 29.07 | 27.25 | 27.73 | **29.20** | 28.83 |
| Fast Food | 26.05 | 26.90 | 24.14 | 24.45 | **28.38** | 27.74 |
| Foreman | 32.22 | **32.74** | 31.82 | 32.10 | 31.70 | 31.52 |
| Stephan | 25.97 | 26.16 | 23.79 | 24.38 | **26.49** | 25.93 |
| Football | 24.36 | 24.93 | 23.24 | 23.01 | **25.11** | 24.25 |

are encoded using H.264 with even frames skipped to generate video bitstreams of 15 fps. The skipped frames are interpolated at the decoder for evaluation. The rate-distortion control function and the RD optimization function are enabled during the encoding process. The averaged bit rates for these test sequences are set to be 384kbps or 512kbps according to the contents.

In order to provide better visual experience, the proposed MV processing method aims to maintain the object structure by adopting the MB merging process. Moreover, the proposed adaptive MCFI scheme attempts to remove ghost artifacts by using the unidirectional interpolation. As a result, our interpolated result may increase the pixel difference comparing to the original frame. In Fig. 4.8, we show a counter example for the conventional quality measures. Although our final results have better visual quality in both (e) and (f), the PSNR performance is worse than the others. Therefore, in additional to using PSNR to measure signal fidelity to its original frame, we also adopt another objective measurement, SSIM, for quality assessment [49], which was also used in [30][26]. Still, this quality measure is not designed for MCFI and cannot truly reflect the real visual experience. However, it provides another quality index from another point of view.

The visual comparisons are presented in Fig. 4.4 to Fig. 4.8. In Fig. 4.4, blockiness can easily be observed in the interpolated results using direct interpolation, VMF, and bidirectional BMA. In Fig. 4.4(e) and Fig. 4.4(f), most of blockiness artifacts are removed by the proposed adaptively weighted MV averaging. Moreover, in Fig. 4.4 (b) to (e), since there is no matched bidirectional predictions for mouth and nose areas from two consecutive decoded frames, a lot of ghost

Table 4.2: SSIM performance comparisons among five frame interpolation methods and the proposed correlation-based MV processing method.

| Sequences | VMF | Selection | Bi-BMA8 | Bi-BMA16 | Bi-Pred | A-Pred |
|---|---|---|---|---|---|---|
| Walk | 0.79 | **0.80** | 0.75 | 0.76 | **0.80** | **0.80** |
| Formula 1 | 0.84 | **0.86** | 0.82 | 0.83 | **0.86** | 0.85 |
| Fast Food | 0.84 | 0.88 | 0.74 | 0.75 | **0.92** | 0.91 |
| Foreman | 0.95 | **0.96** | 0.91 | 0.92 | **0.96** | **0.96** |
| Stephan | 0.88 | 0.89 | 0.76 | 0.78 | **0.90** | 0.89 |
| Football | 0.73 | 0.76 | 0.68 | 0.69 | **0.78** | 0.76 |

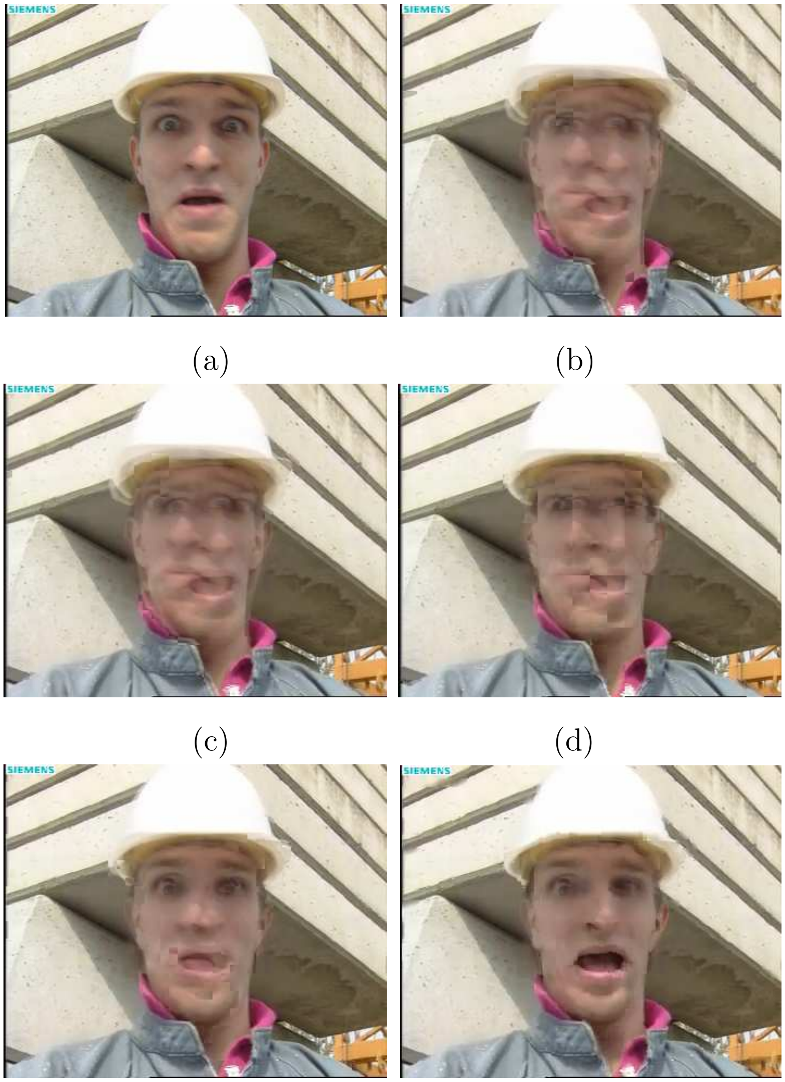
artifacts occur around face areas. By adopting the proposed adaptive frame interpolation scheme, these ghost artifacts are completely removed. Although our interpolated frame is slightly different from the original one, the mouth and the nose areas do follow the motion trajectory from frame 93 to frame 95. You may notice that the PSNR and SSIM performance of the final result is slightly worse than the general bidirectional MCFI scheme. This is because the proposed interpolation scheme attempts to minimize the ghost artifacts in occlusion regions using unidirectional predictions so that the difference between the original frame and interpolated frame increase.

Fig. 4.5 and Fig. 4.6 show examples of using the adaptive frame interpolation scheme. As observed, the ghost effects are greatly reduced in Fig. 4.5(f) and Fig. 4.6(f). The shapes of moving objects are sharper than bidirectional scheme and the boundary between two moving objects looks smoother. Fig. 4.6 shows that even when the frame contains various movements, the proposed adaptive frame interpolation scheme still can perform well and has clearer object contours than other conventional methods. In Fig. 4.7, occlusion occurs in the cross section of the white line and the number area. The proposed MCFI scheme tries to remove ghost artifacts by using unidirectional predictions in this area, so the ghost line does not appear in Fig. 4.7(f). Fig. 4.5 demonstrates an example for the proposed correlation-based MV processing. Since the wall and shirt contain striped textures, the received MVF contains a lot of irregular MVs in these areas. Even when using VMF processing, as shown in Fig. 4.5(c), or correlation based BMA, as shown in Fig. 4.5(d), irregular MVs still cannot be totally removed. By using the proposed correlation-based MV processing method, these irregular MVs are corrected and those dislocation artifacts by using conventional methods do not show up in our interpolated result in Fig. 4.5 and Fig. 4.6(f).

For better comparison, we list averaged PSNR and SSIM values for these six video sequences in Table 4.1 and Table 4.2. As observed, our SSIM performance is consistently better than others but PSNR performance is only better in Walk, Formula 1, and Fast Food. Since the proposed MV processing method includes frame skipping mechanism, the PSNR and SSIM of skipped frames are not counted to the final averaged values for all methods. The frame rates of the interpolated videos are 29.44, 29.18, 28.50, 29.50, 28.30, and 25.50 for Walk, Formula 1, Fast Food, Foreman, Stephan, and Football, respectively.

## 4.5 Acknowledgement

Portions of this chapter appear in “Correlation-Based Motion Vector Processing with Adaptive Interpolation Scheme for Motion Compensated Frame Interpolation,” in *IEEE Transactions on Image Processing*, April 2009, and “Correlation-based Motion Vector Processing For Motion Compensated Frame Interpolation,” in *IEEE International Conference on Image Processing*, Oct. 2008. The dissertation author was the primary author of these publications, and the listed co-author directed and supervised the research that forms the basis for this chapter.

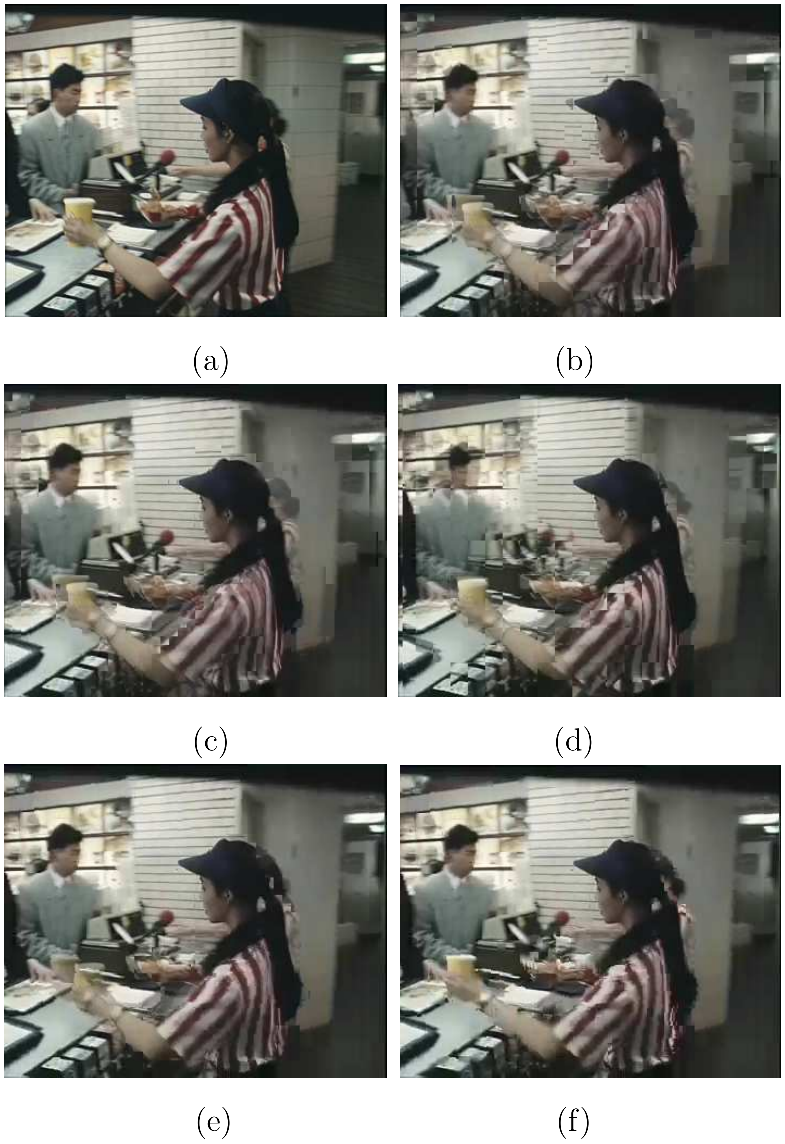


(a) (b)

(c) (d)

(e) (f)

Figure 4.4: The interpolated results of frame 94 of Foreman using (a) original frame, (b) direct interpolation (PSNR: 29.14dB, SSIM: 0.8935), (c) VMF (PSNR: 28.93dB, SSIM: 0.8736), (d) bidirectional BMA (PSNR: 29.19dB, SSIM: 0.8976, (e) the proposed correlation-based MV processing with the bidirectional MCFI scheme (PSNR: 30.71dB, SSIM: 0.9387), and (f) the proposed correlation-based MV processing method with the proposed MCFI scheme (PSNR: 29.75dB, SSIM: 0.9293).

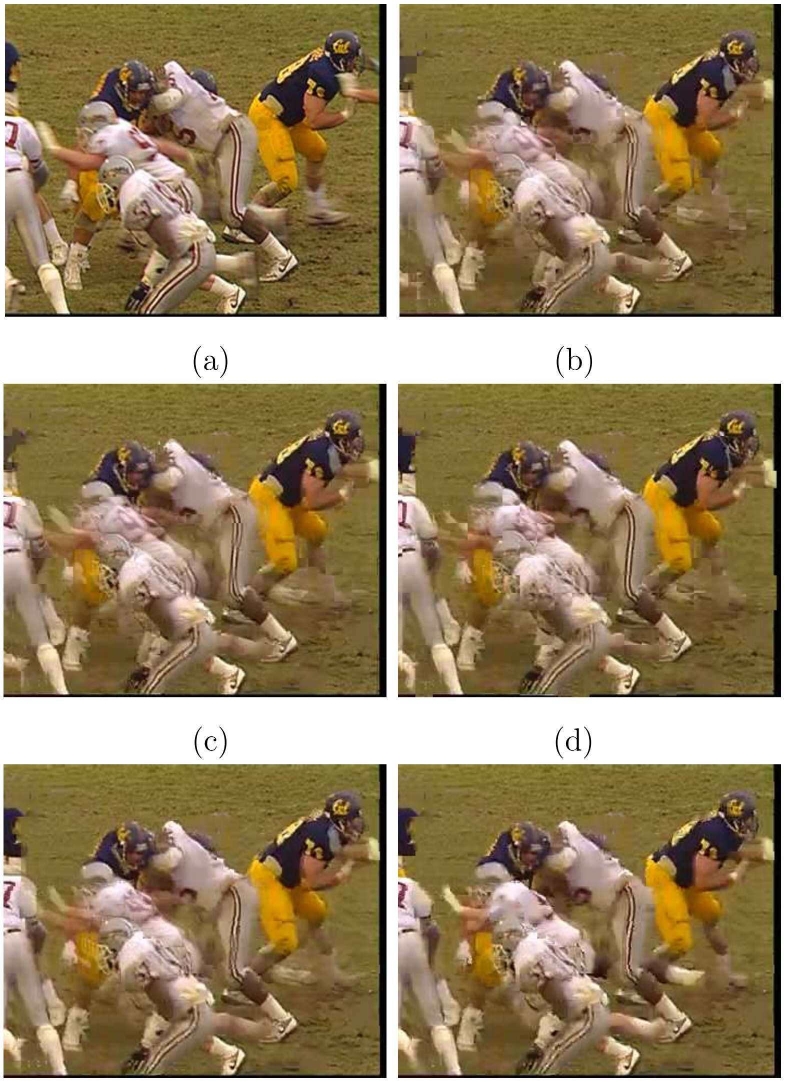


(a) (b)

(c) (d)

(e) (f)

Figure 4.5: The interpolated results of frame 202 of Fast Food using (a) original frame, (b) direct interpolation (PSNR: 22.70dB, SSIM: 0.7918), (c) VMF (PSNR: 22.70dB, SSIM: 0.7915), (d) bidirectional BMA (PSNR: 20.52dB, SSIM: 0.6777, (e) the proposed correlation-based MV processing with the bidirectional MCFI scheme (PSNR: 26.10dB, SSIM: 0.8990), and (f) the proposed correlation-based MV processing method with the proposed MCFI scheme (PSNR: 25.16dB, SSIM: 0.8818).



(a)

(b)

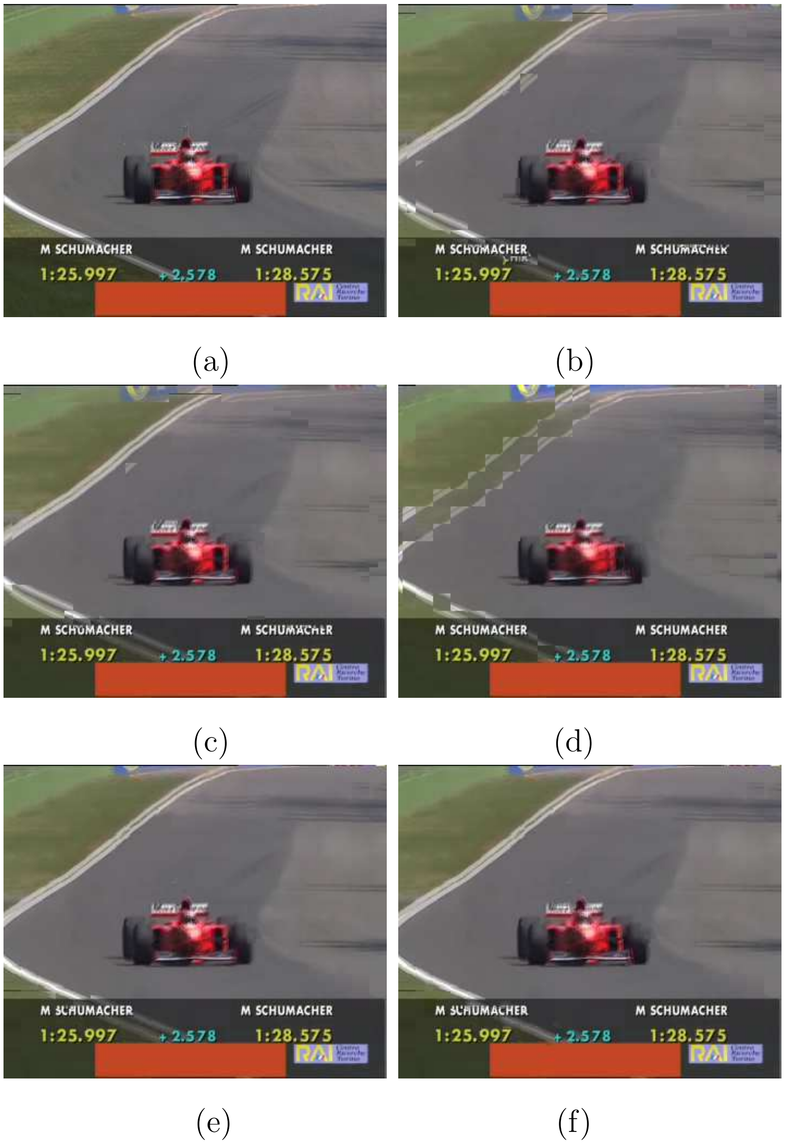
(c)

(d)

(e)

(f)

Figure 4.6: The interpolated results of frame 20 of Football using (a) original frame, (b) direct interpolation (PSNR: 22.60dB, SSIM: 0.6960), (c) VMF (PSNR: 22.80dB, SSIM: 0.7083), (d) bidirectional BMA (PSNR: 22.55dB, SSIM: 0.7237, (e) the proposed correlation-based MV processing with the bidirectional MCFI scheme (PSNR: 22.99dB, SSIM: 0.7405), and (f) the proposed correlation-based MV processing method with the proposed MCFI scheme (PSNR: 22.28dB, SSIM: 0.7172).

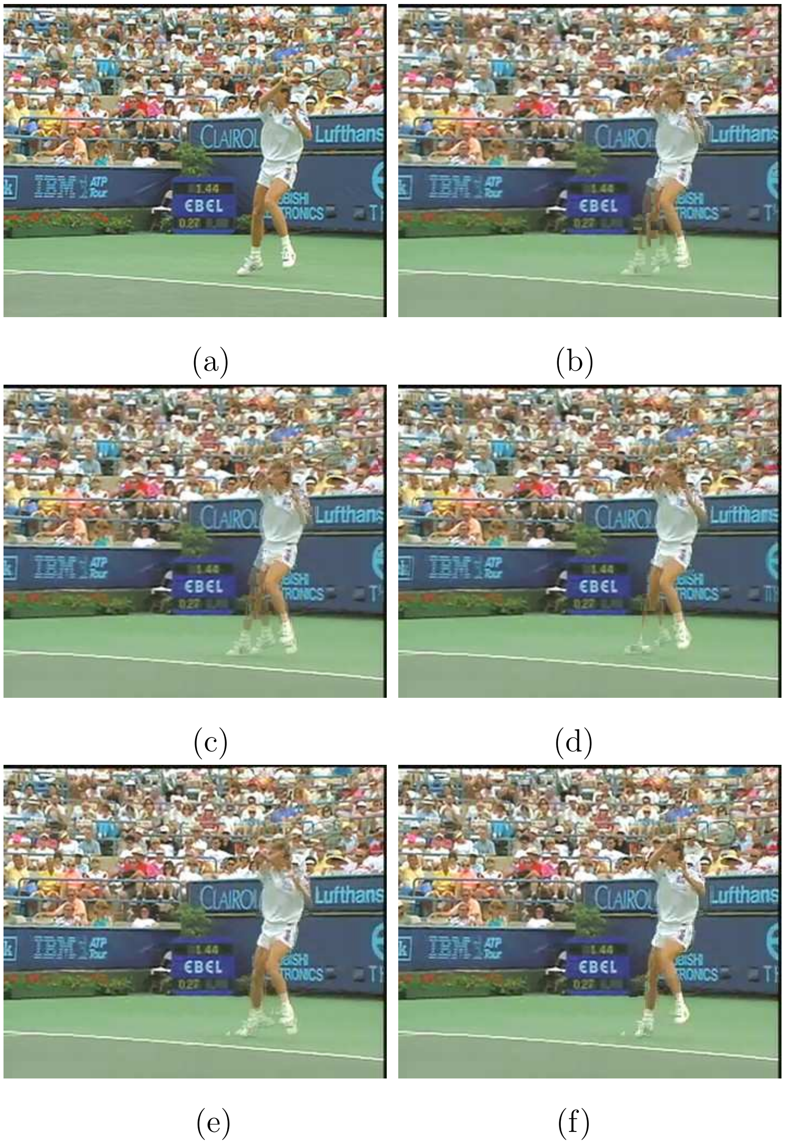


(a) (b)

(c) (d)

(e) (f)

Figure 4.7: The interpolated results of frame 52 of Formula 1 using (a) original frame, (b) direct interpolation (PSNR: 30.53dB, SSIM: 0.9430), (c) VMF (PSNR: 30.50dB, SSIM: 0.9446), (d) bidirectional BMA (PSNR: 26.83dB, SSIM: 0.8278, (e) the proposed correlation-based MV processing with the bidirectional MCFI scheme (PSNR: 33.52dB, SSIM: 0.9681), and (f) the proposed correlation-based MV processing method with the proposed MCFI scheme (PSNR: 32.66dB, SSIM: 0.9632).



(a) (b)

(c) (d)

(e) (f)

Figure 4.8: The interpolated results of frame 112 of Stephan using (a) original frame, (b) direct interpolation (PSNR: 28.28dB, SSIM: 0.9388), (c) VMF (PSNR: 28.50dB, SSIM: 0.9389), (d) bidirectional BMA (PSNR: 28.41dB, SSIM: 0.9445, (e) the proposed correlation-based MV processing with the bidirectional MCFI scheme (PSNR: 27.64dB, SSIM: 0.9351), and (f) the proposed correlation-based MV processing method with the proposed MCFI scheme (PSNR: 26.91dB, SSIM: 0.9255).

# 5 True Motion Processing Based on Motion Trajectory Curve Analysis

In the previous chapters, we performed bidirectional MV selection based on the received MVF, instead of using computationally complex motion re-estimation. Since motion correlation and residual energy distribution are explicitly considered during the bidirectional motion correction process, our proposed method so far can maintain object structures well as compared to conventional MV processing methods. In occlusion regions, the corresponding predictions are adaptively selected based on local motion analysis, and therefore the visual quality of interpolation becomes very sensitive to motion accuracy. To overcome this problem, in this chapter, we further propose using motion temporal reliability as the posterior motion quality check for bidirectional MV selection. By observing BPD values along the temporal axis, we can effectively determine if a MV is suitable for the current interpolated block based on its BPD variation pattern. As a result, the motion accuracy using the proposed algorithm is improved especially in motion boundaries and other areas where the actual movement is difficult to be decided, such as the motion of small objects.

In the following sections, we first address the problems of motion reliability for general motion estimation and motion processing approaches that use SAD or BPD as the major criteria to determine motion in Section 5.1. To solve the reliability problems, the curve analysis derived from temporal BPD variations is

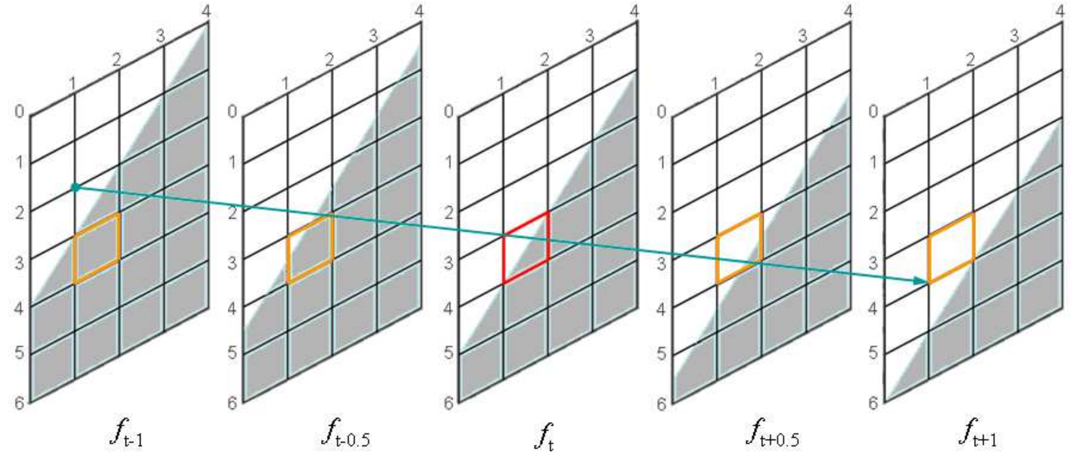


Figure 5.1: Motion compensated frame interpolation. $f_{t-1}$, $f_{t+1}$, and $f_t$ are the previous decoded frame, the current decoded frame and the frame to be interpolated, respectively.

presented in Section 5.2. Details on how to use this curve analysis to assist MV processing is described in Section 5.3.

## 5.1 Motion Reliability Problem for Bidirectional Motion Vector Processing

In Chapter 3 and 4, although the proposed multistage MV processing approach can gradually correct those identified unreliable MVs by minimizing BPD values, it also has the same difficulties as other BPD-based or SAD-based motion estimation methods, such as motion accuracy around occlusion areas as well as areas that have repeated patterns. This is because MVs with minimal BPD values do not always represent the true motion in these areas. Even though the proposed method has explicitly considered motion correlation, if there are too many irregular MVs in the same neighborhood, the interpolated frame may not be perfectly reconstructed. As a result, these temporal incoherent motions can cause motion jerkiness artifacts during video display.

To avoid ambiguous motion matches for occlusion areas or areas that have distorted objects, the work in [19] proposed combining SAD values from the previous frame and the future frame in 3-D recursive search. With constraints on the MV error range, the author found out that the vector error length seems highly correlated to its associated SAD values by observing the spatial SAD distribution in video clips without occlusions. Based on this observation, a SAD probability model was further derived to predict possible MV error. In this model, although SAD values provide a good indicator for motion estimation in most of cases, if the video contains smooth contents, repeat patterns, small moving objects, or occlusions, the SAD distribution may not truly reflect the actual vector error length. Similarly, BPD indices also suffer from the same problems. In these critical areas, image inpainting or image segmentation techniques may help to reduce artifacts, but the high computational complexity requirement limits its usage. Therefore, how to prevent using unreliable motion for these areas is still a challenging problem.

Instead of examining the vector error spatially, we propose analyzing the MV reliability in the temporal domain. In the context of bidirectional scheme for MCFI, we first observe how BPD values distribute along the temporal axis and then analyze if their variations can provide better quality index for MV correction. An ideally interpolated case is shown in Fig. 5.1, where each block size is 8×8. If the motions for both the background and the foreground moving object are correctly estimated, the interpolated frames, $f_{t-0.5}$, $f_t$, and $f_{t+0.5}$, should be perfectly reconstructed at different time scale ratios using the proposed adaptive frame interpolation scheme. In order to observe the BPD variation, we take the block, $b_{4,2}$, as an example, which is denoted by red rectangle in $f_t$ and is denoted by yellow rectangles in other frames. The reason why we choose $b_{4,2}$ for the demonstration is that it is located around the motion boundary where the motion usually cannot be estimated correctly using minimal BPD values.

According to the motion trajectory, we can observe that the temporal co-located blocks, $b_{4,2}$, keep track of the object movement temporally until the gray object disappears. That is, the video contents of the intermediate blocks vary as the gray object moves across the motion boundary. Likewise, the same phenomenon can also be observed in other motion boundary blocks. Based on this observation, we may take advantage of this temporal characteristic of the co-located interpolated blocks to prevent using temporally inconsistent motion for MCFI. In other words, during the MV correction processing, we should only consider the MV that is capable of illustrating possibly right motion traces so that ambiguous motion can be avoided around occlusions and motion boundaries. Since we adaptively select motion compensations for occlusion areas, the performance of this interpolation scheme can also be promoted and have better visual quality.

## 5.2 Motion Vector Temporal Reliability Analysis

In the previously proposed MCFI method, the major problem for bidirectional MV processing approaches is that the selected MVs with minimal $ABPD$ values may not represent the actual motion. Therefore, we can often observe broken moving structures when the background has relatively smooth contents. This phenomenon is more obvious when the moving object is small or does not have strong edges. To prevent the occurrence of this kind of inconsistent motion, an additional motion quality examination should be performed to check if the selected MV can truly match the actual motion flow. Based on the previous discussion, the correct motion trajectory could be predicted by observing the changes of video content among co-located bidirectional interpolated blocks at different time scale ratios. The simplest way to achieve this is to calculate $ABPD$ values using several combinations of forward and backward MVs according to the corresponding temporal positions and further analyze how these $ABPD$ values distribute temporally. If $ABPD$ variations do not deviate from the predefined movement patterns, which we will describe in greater details later, this MV will more likely be the correct motion for the currently investigated block.

### 5.2.1 Motion Vector Temporal Curve Derivation

In Eqn. (3.3), the block-based $ABPD$ values for the middle interpolated frame is calculated by dividing the MVs by two to form the forward MV and the backward MV so that the bidirectional predictions can be obtained from $f_{t-1}$ and $f_{t+1}$ accordingly. In order to observe temporal $ABPD$ variations, the temporal

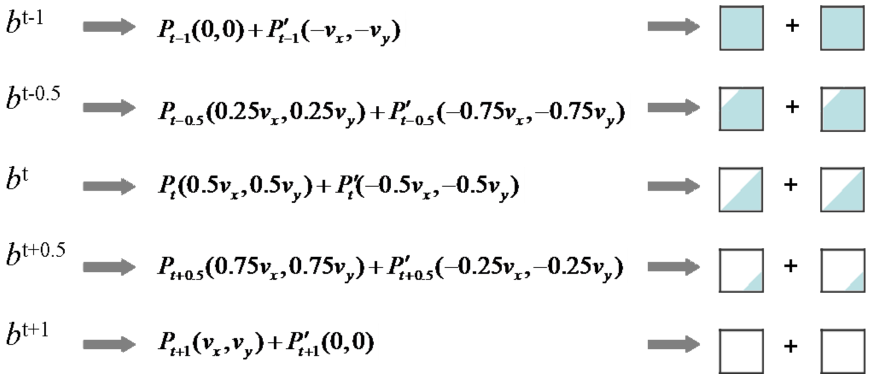

$b^{t-1}$
$P_{t-1}(0,0)+P'_{t-1}(-v_x,-v_y)$
+
$b^{t-0.5}$
$P_{t-0.5}(0.25v_x,0.25v_y)+P'_{t-0.5}(-0.75v_x,-0.75v_y)$
+
$b^{t}$
$P_t(0.5v_x,0.5v_y)+P'_t(-0.5v_x,-0.5v_y)$
+
$b^{t+0.5}$
$P_{t+0.5}(0.75v_x,0.75v_y)+P'_{t+0.5}(-0.25v_x,-0.25v_y)$
+
$b^{t+1}$
$P_{t+1}(v_x,v_y)+P'_{t+1}(0,0)$
+

Figure 5.2: The individual prediction at each time label. $\mathbf{v} = (v_x, v_y)$ is the examined MV, and $P(\mathbf{v}_{fw})$ and $P'(\mathbf{v}_{bw})$ represent the bidirectional predictions using forward and backward MVs, respectively.

axis is first equally divided by five and their corresponding $ABPD$ values can be obtained using similar manners. That is, different proportions of forward and backward motions will be employed to fetch the predictions at each time point.

For all temporal co-located blocks, an examined MV will be multiplied by five different scale ratios to form different forward and backward MV sets. Their derived prediction sets will be used for the subsequent $ABPD$ calculation. Hence, Eqn. (3.3) can be modified as follows:

$$ABPD_{\mathbf{v}}(k) = \sum_{x,y \in b^k_{m,n}} |f_{t-1}(x+\frac{k}{4}v_x, y+\frac{k}{4}v_y) - f_{t+1}(x-\frac{4-k}{4}v_x, y-\frac{4-k}{4}v_y)| \quad (5.1)$$

where $k = \{0, 1, 2, 3, 4\}$ can also be used to represent the time labels $\{t-1, t-0.5, t, t+0.5, t+1\}$, respectively. $b^k_{m,n}$ denote the temporally co-located blocks.

We again use Fig. 5.1 to further illustrate this process. Considering the gray object motion $(0, -16)$ as the most appropriate MV for $b^2_{4,2}$, all the combinations of the forward and backward MV sets and their corresponding predictions can therefore be obtained as shown in Fig. 5.2. The forward and backward prediction sets, which are respectively denoted as $\{P_{t-1}, P_{t-0.5}, P_t, P_{t+0.5}, P_{t+1}\}$ and $\{P'_{t-1}, P'_{t-0.5}, P'_t, P'_{t+0.5}, P'_{t+1}\}$, are obtained by multiplying the MV with different forward and backward scale ratios. For example, $b^4_{4,2}$ is the averaged result using the forward prediction and the backward prediction obtained by $(0.75v_x, 0.75v_y)$ and $(-0.25v_x, -0.25v_y)$. In Fig. 5.2, we can observe that the portion of the gray

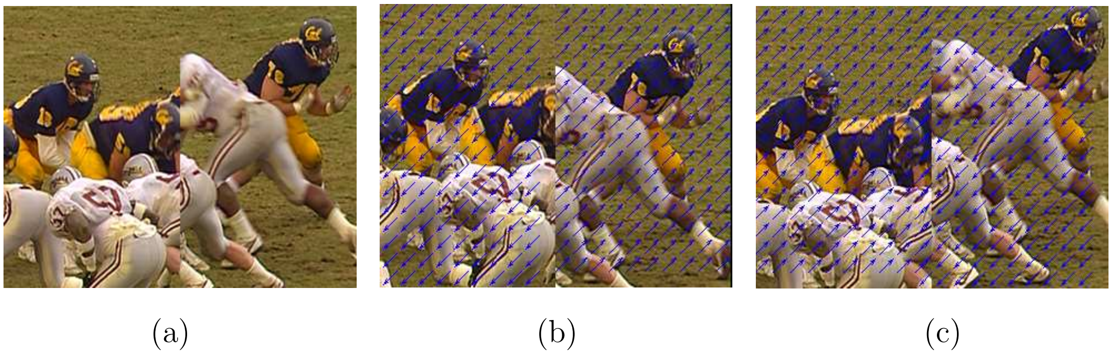


Figure 5.3: Synthetic images of Football. (a) Left side is frame 1 and right side is frame 15. (b) Left side is shifted with motion $(-12, 12)$ and right side is shifted with motion $(12, -12)$. (c) Left side is shifted with motion $(12, -12)$ and right side is shifted with motion $(-12, 12)$.

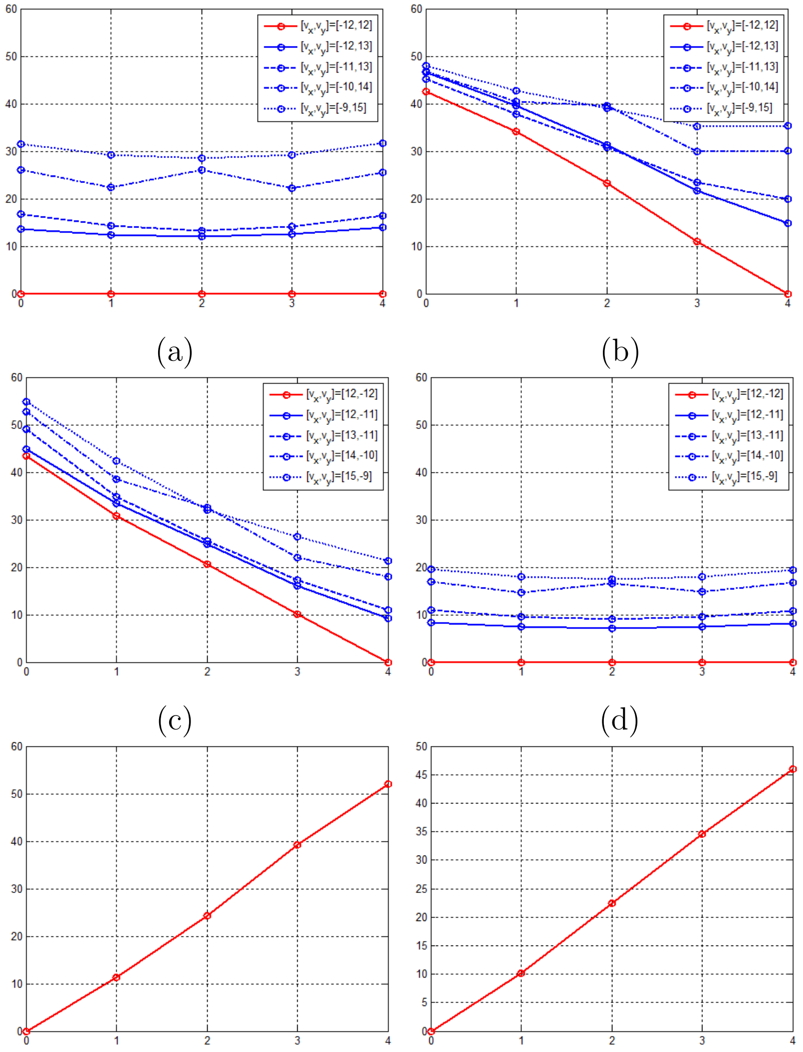
object decreases from the initial of the motion trajectory to the end of the motion trajectory. That is, the corresponding $ABPD$ values should increase in the same direction since $(0, -16)$ is not the motion for the white background. In practice, the white background usually has dissimilar motion and is composed of more complex scenes. The $ABPD$ values should become relatively larger as the major portion of the predictions has become different moving objects. Likewise, if the gray object moves in the opposite direction, i.e., $(0, 16)$, the corresponding $ABPD$ values may have reverse variations. According to this observation, the resulting $ABPD$ variations seem highly correlated to the directions of the given MVs and the neighboring motion distribution. Therefore, we should further take advantage of the temporal $ABPD$ variation curvature to analyze the motion reliability so that the temporally incoherent motion can be detected.

### 5.2.2 Motion Vector Temporal Curve Statistics

Based on the assumption that possibly correct MVs should have regular patterns of $ABPD$ value variations, we would like to analyze what kinds of $ABPD$ curvatures have higher possibility to be the right motion so that these patterns can be further classified for temporal motion quality check. Since correct motions in video clips cannot be easily obtained, except for panning or zooming/out sequences, we use a synthetic images by combining frame 1 and frame 15 of FOOTBALL as shown in Fig. 5.3. Fig. 5.3(a) represents the previous frame, and Fig. 5.3(b) and Fig. 5.3(c) represent the current frames with reverse motions. The statistics of $ABPD$ variations for Fig. 5.3(b) are illustrated in Fig. 5.4(a)-(d). The $ABPD$ values are calculated based on block size of 16×16 and their statistic information is obtained by partitioning Fig. 5.3(b) into four parts: left panning, left motion boundary (the left MB column from the middle), right motion boundary (the right MB column from the middle), and right panning. Their conclusive $ABPD$ curves are obtained by mean manner as shown in Fig. 5.4.

In Fig. 5.4, we can observe that the curvatures tend to have flat shapes for panning portions ((a) and (d)) and left-ascending shapes for the motion boundary portions ((b) and (c)), in which curvatures of correct motion are denoted using red color and curvatures of deviated motion are denoted using other colors. The flat curvature matches our expectation since the panning motion is the best motion for every co-located block. As a result, the $ABPD$ values of the correct MV in the panning portions are zeros at all time labels. In Fig. 5.4(b), since the examined motion, $(-12, 12)$, originates from the right motion boundary portion and points to the left motion boundary portion, the highest $ABPD$ value occurs at $k = 0$ (origin). As the obtained bidirectional motion compensations become closer to the left motion boundary region, the $ABPD$ values decrease accordingly and the smallest value occurs at $k = 4$. Similarly, the MV of the right motion boundary portion also originates from areas with different motion, and its $ABPD$ curvature has the same variation patterns as shown in Fig. 5.4(c). In order to investigate the sensitivity of $ABPD$ variation patterns, we allow small deviations of the correct MVs to check if the curvatures can still be maintained well. As shown in Fig. 5.4(a)-(d), the blue curves still follow the same directions as red curves and, obviously, the $ABPD$ difference decreases as motion deviations become less.

To further verify our argument, we create another synthetic image with reverse motion in Fig. 5.3(c) and the resulting $ABPD$ curves for the left motion boundary portion and the right motion boundary portion are demonstrated in Fig. 5.4(e) and (f), respectively. As expected, the curvatures have reverse directions

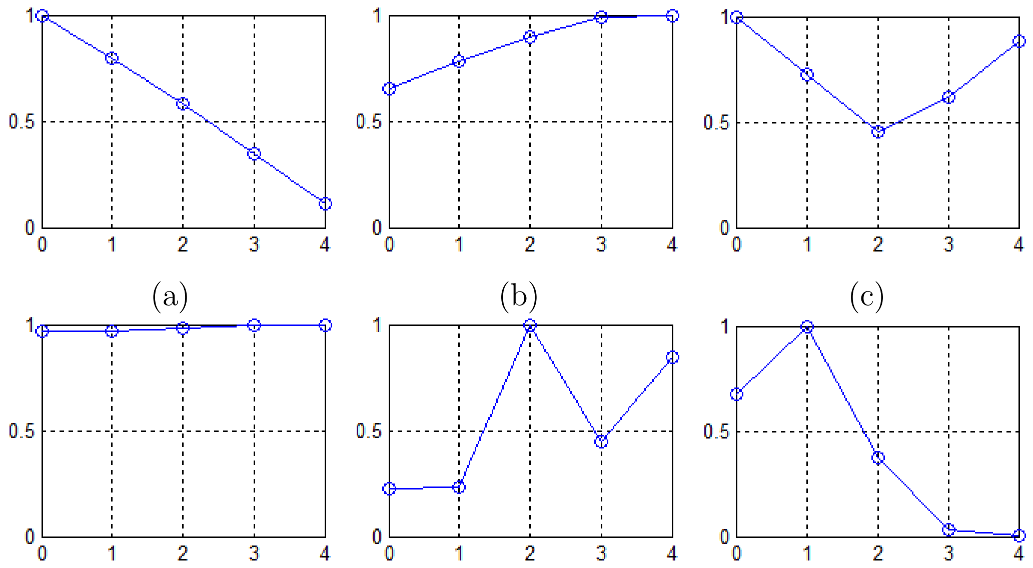


(a) (b)

(c) (d)

(e) (f)

Figure 5.4: Statistical $ABPD$ variations. The y axis and the x axis represent the averaged values of $ABPD_{\mathbf{v}}(k)$ and index $k$, respectively. (a), (b), (c), and (d) are the statistical $ABPD$ curves using correct MV for left panning, left motion boundary, right motion boundary, and right panning portions, respectively. (e) and (f) are $ABPD$ curves for left motion boundary and right motion boundary portions, respectively in Fig. 5.3(c).



(a) (b) (c)

(d) (e) (f)

Figure 5.5: Examples of *ABPD* curves. The y axis and the x axis represent the normalized $ABPD_{\mathbf{v}}(k)$ and index $k$, respectively.

as compared to Fig. 5.4(b) and (c), since their MVs point to the areas that have dissimilar motions.

In Fig. 5.5, we demonstrate several typical temporal *ABPD* variations that are obtained from actual video clips. For Fig. 5.5(a) and (b), either around the origin or the end of the MV should have similar motions. If the size of the moving object is very small, its corresponding *ABPD* curve and Fig. 5.5(c) should be alike. In addition to the possible curves around the motion boundary, the areas inside the object or panning areas should have small *ABPD* variations such as Fig. 5.5(d). We also demonstrate some examples for non-regular *ABPD* curves resulting by incorrect MVs in Fig. 5.5(e) and (f), which seems to vary randomly along motion trajectory. This phenomenon often occurs when a moving object is very small, such as head and legs, and dissimilar background motion is used to replace the object's actual movement.

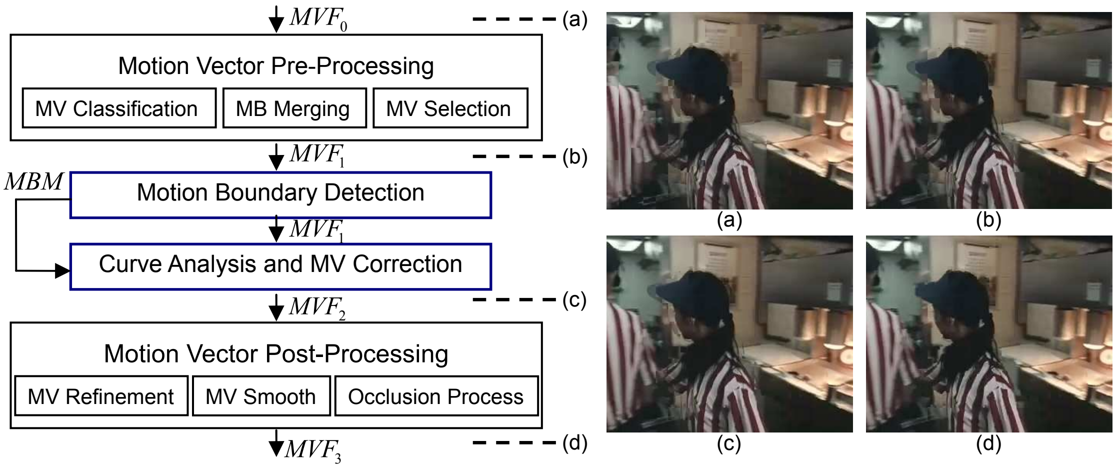


Figure 5.6: The flowchart of the proposed MV processing method. (a) is the interpolated result using the received MVF. (b) is the interpolated result after the MV pre-processing stage. (c) is the interpolated result after the proposed temporal motion analysis and correction procedure. (d) is the final interpolated result using the adaptive MCFI scheme.

## 5.3 Motion Vector Processing with Temporal Reliability Analysis

In order to improve the MV reliability for MCFI, we take the proposed correlation-based MV processing approach described in Chapter 4 as the baseline with further consideration on motion temporal reliability. That is, based on the temporal characteristics of $ABPD$ variations, we propose using curve analysis as a motion quality check to identify inappropriate motion that has small $ABPD$ value for $f_t$, but do not follow the actual motion flow. In Fig. 5.4 and Fig. 5.5, if we take $ABPD_{\mathbf{v}}(2)$ as the curve center, which is also the $ABPD$ value for $f_t$, we can observe that most of the correct curve shapes can be categorized into left ascending, right ascending, bilateral ascending, and flat curves. Hence, if we cannot find matched curvatures in any of curve types described above, this MV is more likely to deviate from the correct motion flow.

Since we employ the codec-based MVF where MVs are often unavailable due to intra-coded mode, instead of verifying the MV temporal reliability for the MV candidate sets in Eqn. (3.3), we perform the MV $ABPD$ curve analysis when most of MVs have been determined. Therefore, we partition the correlation-based MV processing into two parts, MV pre-processing and MV post-processing as shown in Fig. 5.6. The first stage includes the MV reliability classification, the MB merging process, and the bidirectional MV selection process. As most of the MVs have been corrected, we only perform the temporal curve analysis for motion boundary regions. This is because $ABPD$ curvatures are supposedly to diverge when the surrounding motion distribution starts becoming different. The curve check can assist in rearranging the motion distribution around occlusion areas and furthermore ambiguous motion can be corrected if the previously selected MV is incoherent to neighboring MVs. Based on the rearranged MVF, the MV refinement by analyzing the $ABPD$ energy distribution, the MVF interpolation, and the occlusion process will be performed subsequently in the MV post-processing stage.

### 5.3.1 Motion Boundary Detection

Motion boundaries are the locations where motion starts to differ, and we can simply use MV magnitude distance as the criterion to detect possible motion boundary areas. For each MV with the block size of 8×8, we calculate its motion magnitude distance between adjacent MVs and use the maximal distance as the index to determine if the current block is located around motion boundaries. The distance index for each MV can be represented as follows:

$$d_{m,n} = \underset{i,j=\{-1,0,1\}}{\arg\max}\,(||\mathbf{v}_{m,n} - \mathbf{v}_{m+i,n+j}||_2). \tag{5.2}$$

According to the distance index, the motion boundary map can be defined as follows:

$$MBM_{m,n} = \begin{cases} 1, & \text{if } d_{m+i.n+j} > \alpha_d \text{ for } i,j = \{-1,0,1\}, \\ 0, & \text{otherwise.} \end{cases} \tag{5.3}$$

In this map, not only will the center distance index be considered for the classification process, but we also consider the neighboring distance indices. Once any of the distance indices are greater than the predefined threshold, $\alpha_d$, the current block will be classified as a motion boundary block. The reason why we check the neighboring blocks is that if MVs have larger magnitude, the range of the motion boundary region should become relatively large as well. In the following MV temporal reliability check, only MBs having at least one motion boundary block will be further analyzed if the previously selected MV can truly represent the temporal motion trace.

### 5.3.2 Motion Correction Based on Curve Analysis

In the raster scan order, each MB will be checked if it contains motion boundary MVs. In this case, the $ABPD$ curve analysis will be performed at both the previously selected MVs and adjacent MVs to further verify which MVs can better represent the temporal motion trace. During the temporal reliability classification, the resulting $ABPD$ curvature types of these MV candidates will be first identified. In the proposed MV temporal reliability analysis, sometimes it is very difficult to find MVs that can match the curvature pattern perfectly, especially for intra-coded MBs. In order to reduce the sensitivity of motion accuracy, we normalize the temporal $ABPD$ curve and use a 2-tap lowpass filter to smooth the curvature, which can be represented as follows:

$$\Phi_{\mathbf{v}}(n) = \sum_{k} h(n-k) \frac{ABPD_{\mathbf{v}}(k)}{ABPD_{\mathbf{v},max}} \tag{5.4}$$

where

$$ABPD_{\mathbf{v},max} = \arg\max(ABPD_{\mathbf{v}}(k)). \tag{5.5}$$

MVs will be regarded as reliable motion if $\Phi_{\mathbf{v}}$ can find similar curvature from a predefined curve set. On the contrary, if the matched variation patterns cannot be found for $\Phi_{\mathbf{v}}$, the MV will be classified as unreliable. For those MVs that have very small magnitude so that their curvatures usually have flat variations, we simply classify these MVs as temporally reliable without curvature comparisons. Moreover, since $ABPD$ temporal variation only provides the information about possible movement of the moving object, the correctness of its curvature actually depends upon how the surrounding motion distributes. Hence, according to the obtained curvature, we should further analyze if the blocks where the examined MV originates and the blocks where the examined MV ends have similar or dissimilar MVs.

If the $ABPD$ variation belongs to the left-ascending curvature type, we will further check if the blocks where the examined MV points to have similar MVs. If positive, the examined MV is more likely to fit in with the adjacent motion distribution and will be considered to replace the motion of the current MB. On the other hand, if the $ABPD$ variation matches the right-ascending curvature, the MVs of the blocks where the examined MV originates from will be checked to ensure its correctness. For MVs whose curves are classified as flat-curvature, neighboring MVs from both sides will be investigated. In such a case, if similar MVs can be found from either side, the MV should be coherent to the surrounding motion distribution. For the bilateral ascending curvature, since there is no clear sign of indicating what kind of neighboring motion distribution should be, we consider all MVs having this type of curvature as temporal reliable motion. If we define the previously selected MVs and the adjacent MVs as a MV candidate set, $S$, this MV temporal reliability classification process ($MVTR$) for each MV within $S$ can be written as follows:

$$MVTR_{\mathbf{v}} = \begin{cases} 1, & \text{if } \Phi_{\mathbf{v}} \in c_B, \\ 2, & \text{if } \Phi_{\mathbf{v}} \in c_L \text{ and } \arg\min\limits_{\mathbf{v_n} \in S_2}(||\mathbf{v}-\mathbf{v_n}||_2) < \alpha_d, \\ 3, & \text{if } \Phi_{\mathbf{v}} \in c_R \text{ and } \arg\min\limits_{\mathbf{v_n} \in S_1}(||\mathbf{v}-\mathbf{v_n}||_2) < \alpha_d, \\ 4, & \text{if } \Phi_{\mathbf{v}} \in c_F \text{ or } ||\mathbf{v}||_2 < \alpha_m, \text{ and} \\ & \qquad \arg\min\limits_{\mathbf{v_n} \in \{S_1, S_2\}}(||\mathbf{v}-\mathbf{v_n}||_2) < \alpha_d, \\ 0, & \text{otherwise.} \end{cases} \tag{5.6}$$

Here, $c_B, c_L, c_R, c_F$ are the predefined curvatures for bilateral ascending, left ascending, right ascending, and flat curves, respectively. $\alpha_m$ is the threshold value for MV magnitude and we simply set it with value of 4 to avoid half-pixel or quad-pixel $ABPD$ calculation. For each examined MV, $S_1$ and $S_2$ represent the neighboring MVs obtained from the reverse direction and the same direction, respectively. That is, for a 3×3 MV matrix, if the x component of the center MV is less than zero and the y component is greater than zero, $S_1$ consists of the MVs from the right column and the upper row, while $S_2$ consists of the MVs from the left column and the bottom row.

In the temporal reliability classification, only the MVs whose curvatures matched both the predefined patterns and the requirement of neighboring motion distribution will be chosen for the current MB. However, if there is no reliable MVs available in this neighborhood after the sorting process, we skip the MV rearrangement process for the current MB. The MVs that have passed the temporal quality check can be therefore represented as follows:

$$S' = \begin{cases} \mathbf{v}, & \text{if } MVTR(\mathbf{v}) > 0 \text{ and } \mathbf{v} \in S, \\ null, & \text{otherwise.} \end{cases}$$

Among all qualified MVs, the best MV for the current interpolated MB can be selected based on the minimal $ABPD$ values. This bidirectional selection process can be defined as follows:

$$\mathbf{v}_b^* = \arg\min_{\mathbf{v} \in S'} (ABPD_{\mathbf{v}}(2)).$$

This MV correction procedure is regarded as the temporal reliability verification process for motion boundary areas, and the computational complexity only slightly increases. The new obtained MVF will be further refined (8×8) and be interpolated (4×4) in the subsequent stages for MCFI. In Fig. 5.6, we also demonstrate how the image quality be improved using the temporal motion reliability analysis and MV correction. We can observe that the motion boundary MVs are corrected around the cap, shirts, and face areas as compared to Fig. 5.6(b) and (c). Although the visual artifacts occur in the bidirectional MCFI result as shown in Fig. 5.6(c), these artifacts are removed in Fig. 5.6(d) since the predictions around motion boundary areas are adaptively selected based on local motion distribution.

## 5.4 Simulations

We compare the proposed method with original non-skipped frames, direct MCFI, and the previously proposed method in Chapter 4. Eight video sequences of CIF size, Bus, Fast Food, Football, Foreman, Kayak, Stephan, and Walk are used to evaluate the performance of the proposed temporal motion

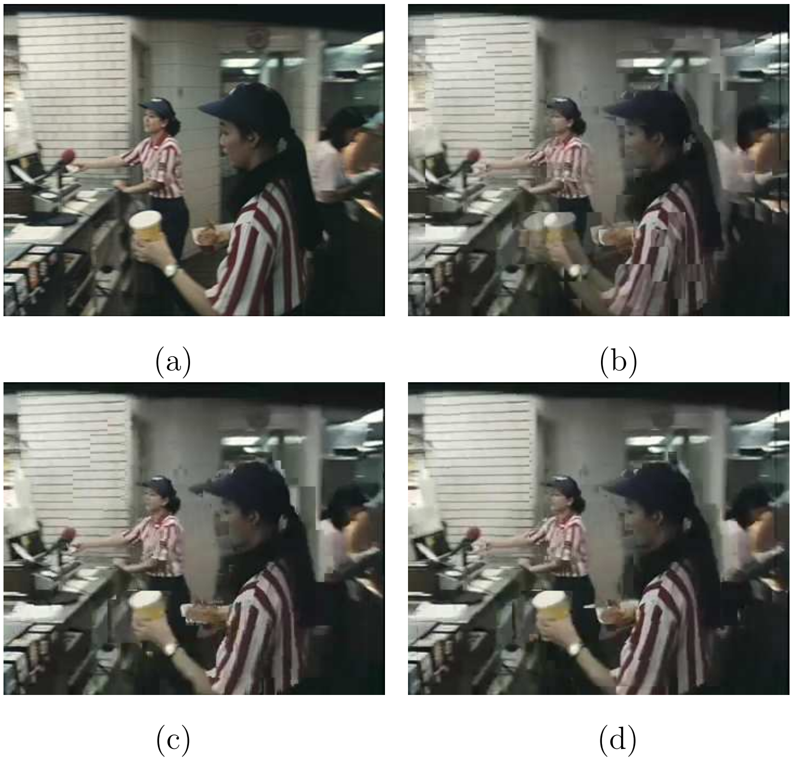


Figure 5.7: The interpolated results of frame 192 of Fast Food using (a) original frame, (b) direct MCFI (PSNR: 23.70dB, SSIM: 0.8035), (c) The method in [4] (PSNR: 27.12dB, SSIM: 0.8925), and (d) the proposed method (PSNR: 27.53dB, SSIM: 0.9257).

curve analysis. These video sequences are encoded using H.264 at the bit-rate of 384/512kbps with even frames skipped. These skipped frames are then interpolated at the decoder for evaluation.

The visual comparisons are illustrated in Fig. 5.7 to Fig. 5.9. In Fig. 5.7(b), we can observe that the received MVF contains many irregular MVs and intra-coded MBs. Although the previously proposed method is able to correct most of unreliable MVs, visual artifacts still occur around the wall areas and motion boundary regions as shown in Fig. 5.7(c). However, these artifacts can be removed using the proposed method as shown in Fig. 5.7(d), since the temporal motion analysis can find more reliable motion when the local motion distribution starts to differ. Moreover, the motion quality between the foreground moving object and the background is improved so that the artifacts around the motion boundaries,

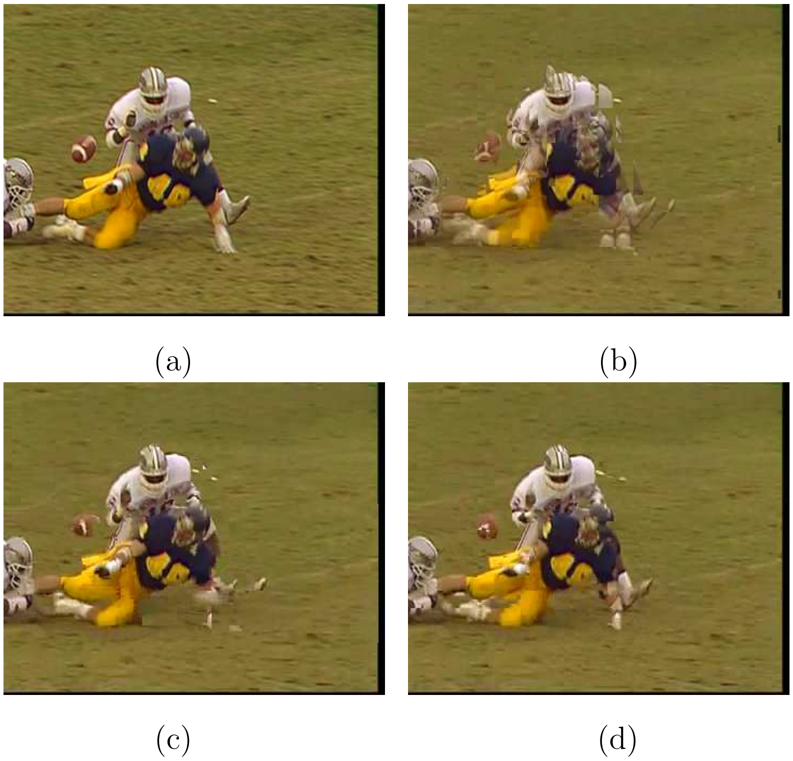
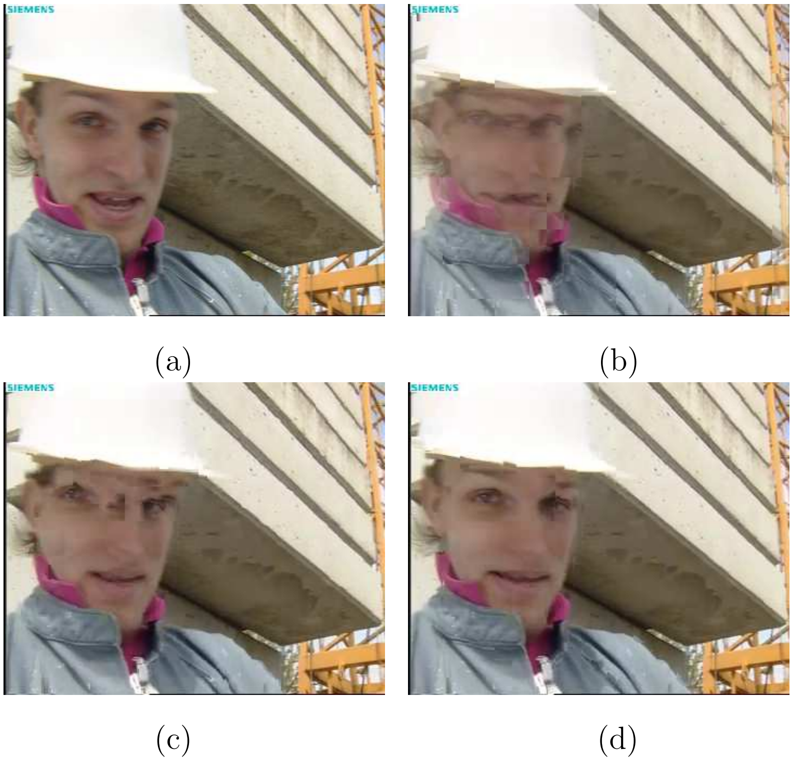


Figure 5.8: The interpolated results of frame 192 of FOOTBALL using (a) original frame, (b) direct MCFI (PSNR: 25.60dB, SSIM: 0.7451), (c) The method in [4] (PSNR: 25.81dB, SSIM: 0.7654), and (d) the proposed method (PSNR: 26.88dB, SSIM: 0.7874).

such as the cap, hair, and hand areas, are reduced as well.

In Fig. 5.8, the bidirectional MV selection process merely based on minimal $ABPD$ values cannot assign correct motion to motion boundary regions especially when the background has relatively smooth contents or the motion has more complex distribution. From Fig. 5.8(b) to (c), we can observe that the motion of the major moving structures has been corrected using the previously proposed correlation-based MV processing method. However, the visual quality is further improved in Fig. 5.8(d) such as in the leg area. This is because the motion around the motion boundary has been modified. In Fig. 5.9(c), we demonstrate the difficulty to find the correct motion for the distorted moving objects by only considering minimal $ABPD$ values and spatial motion correlation. As shown in Fig. 5.9 (d), the temporal $ABPD$ variation analysis can assist us to select better motion for



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

Figure 5.9: The interpolated results of frame 174 of Foreman using (a) original frame, (b) direct MCFI (PSNR: 27.33dB, SSIM:0.8598), (c) The method in [4] (PSNR: 29.32dB, SSIM: 0.9212, and (d) the proposed method (PSNR: 31.00dB, SSIM: 0.9449).

the distorted face areas.

The averaged PSNR and SSIM results are presented in Table 5.1 and Table 5.2, respectively. As observed, the proposed method outperforms other methods in PSNR and SSIM comparison. One may notice that the average quality index difference between the proposed method and the previous method only shows little improvement as compared to direct MCFI. This is because the proposed MV processing method focuses on the areas having motion boundaries, and these improvements may not be observed from the averaged values.

Table 5.1: PSNR performance comparisons among two frame interpolation methods and the proposed method for eight video sequences.

| Sequences | Direct | Previous | Proposed |
|---|---|---|---|
| Bus | 25.98 | 25.98 | **26.10** |
| Fast Food | 25.78 | 28.24 | **28.39** |
| Football | 23.79 | 24.51 | **24.82** |
| Foreman | **32.03** | 31.55 | 32.00 |
| Formula 1 | 28.47 | 29.14 | **29.17** |
| Kayak | 27.76 | 28.45 | **28.56** |
| Stephan | 25.28 | **25.88** | **25.88** |
| Walk | 22.86 | 23.16 | **23.29** |

Table 5.2: SSIM performance comparisons among two frame interpolation methods and the proposed method for eight video sequences.

| Sequences | Direct | Previous | Proposed |
|---|---|---|---|
| Bus | **0.91** | **0.91** | **0.91** |
| Fast Food | 0.84 | **0.92** | **0.92** |
| Football | 0.69 | 0.74 | **0.76** |
| Foreman | 0.94 | **0.95** | **0.95** |
| Formula 1 | 0.84 | **0.86** | **0.86** |
| Kayak | 0.87 | **0.89** | **0.89** |
| Stephan | 0.84 | 0.86 | **0.87** |
| Walk | 0.79 | **0.81** | **0.81** |

## 5.5 Acknowledgement

Portions of this chapter appear in “Correlation-Based Motion Vector Processing with Adaptive Interpolation Scheme for Motion Compensated Frame Interpolation,” in *IEEE Transactions on Image Processing*, April 2009, and “Correlation-based Motion Vector Processing For Motion Compensated Frame Interpolation,” in *IEEE International Conference on Image Processing*, Oct. 2008. The dissertation author was the primary author of these publications, and the listed co-author directed and supervised the research that forms the basis for this chapter.

# 6 Conclusions

Frame interpolation that uses motion information in the received bitstream is a very simple yet effective technique for compressed video to improve temporal quality by increasing frame rate at the decoder. However, not all the received motion information is suitable for frame interpolation as block-based motion estimation at the encoder often fails to find true motion.

In Chapter 3, we have shown that those unreliable MVs can be identified by their prediction residual energies. We further presented a hierarchical MV processing algorithm based on the classified MV reliability information to produce a more reliable MVF for frame interpolation. Our method analyzes the distribution of high residual energies to effectively merge MBs that are on the motion boundaries. Each merged group is first assigned a single motion and further be hierarchically refined by gradually reducing block size. In this way, complicated object-based segmentation can be avoided to maintain object structure information. Throughout the proposed MV processing method, chrominance information is explicitly considered as it provides valuable information to identify and correct unreliable MVs especially for object edges.

The proposed multi-stage MV processing scheme can work with complicated MVF and complex texture, and more importantly, structure information is better preserved. It is also a low-complexity, standard compliant solution at the decoder since we only perform MV processing to accomplish the concept similar to object-based frame interpolation but without actual edge detection or motion estimation. However, not all of the unreliable MVs can be detected using the received residual energies, such as low-correlated MVs. To correct motion even in smooth or repeat pattern areas, we proposed a correlation-based MV processing method based on the existing work.

In Chapter 4, we assume that those similar irregular MVs should belong to the same object and therefore create another MB merging map for identified low-correlated MVs. During the proposed correlation-based MV selection process, not only can more pixels be referenced for MV selection, but motion correlation is explicitly considered to find more suitable MV candidates. This motion correlation information is further employed to improve the motion smoothing process for the proposed multi-stage scheme. Moreover, to solve general occlusion issues, the block-based motion distribution around the occlusion areas is analyzed to determine appropriate unidirectional predictions, i.e. uncovered or covered regions. As a result, the interpolated results using the proposed method can remove most of the ghost artifacts and obtain clearer object contours. After all, SSIM indexes and visual comparisons also show that our MV processing method outperforms other conventional methods.

Independent to the previously proposed work, we present a novel MV processing approach by analyzing temporal ABPD variations in Chapter 5. Based on this analysis, the MV reliability can actually be predicted by surrounding motion distribution and the corresponding ABPD variation curvature. From the experimental results, this proposed method is very robust in correcting unreliable motion especially in the motion boundaries, areas having repeat patterns, and areas having small moving objects. Therefore, to further improve the motion accuracy for multi-stage MV processing, we employed this motion temporal analysis as a posterior quality check for the bidirectional MV selection procedure. Consequently, the subsequent adaptive frame interpolation scheme can effectively select the suitable predictions for occlusion areas. This motion quality check is considered in motion boundary regions since incorrect MV selection results can only occur when areas contain deviated motions. With additional temporal MV reliability analysis, simulation results also show the proposed method outperforms others in terms of visual quality. Please note that this work is not limited only to bidirectional MV processing methods, since its temporal motion curvature analysis can be extended to block-based motion estimation. Hence, the proposed method can also be extended to assist the accuracy of general motion estimation methods.

In this thesis, the proposed MV processing methods have demonstrated its effectiveness in correcting the bitstream motion at the decoder. Although we only consider the skipping factor of two in our experiments, it can be easily extended to different factors by modifying the MV ratios during bidirectional MV selection. In addition, because the proposed method is designed to analyze motion vectors, it can also be applied during the encoding process. That is, analyzing the difficulties of reconstructing skipped frames based on the bitstream motion so that the skipping factor can be adaptively adjusted. For example, if video clips contain smooth motion such as panning motion, more frames can be dropped and they can still be well reconstructed at the decoder. This idea not only can provide flexibility for MCFI, but also can elaborate the frame interpolation techniques to further improve the coding efficiency.

In the proposed algorithms, the characteristics of color information, residual energy distribution, motion correlation, and temporal motion reliability are explicitly considered to assist MV processing. We can also incorporate these concepts for other motion estimation approaches to promote the motion accuracy. This can be achieved by considering the proposed method as the posterior MV examination, using motion correlation and temporal reliability to eliminate unreliable MV candidates, or taking color elements in conjunction with MV search to strengthen the object edge information.

In addition to the applications in motion estimation and frame interpolation, the proposed method can also be used in other video applications that need accurate motion field such as video transcoding and error concealment. To reduce complexity from motion re-estimation in the transcoder, the proposed MV processing method can be used to convert the MVF into different formats (block size). This is because we analyze the motion distribution based on the residual energy and hierarchically correct possibly unreliable MVs. For the error concealment application, the proposed method can be used to recover the corrupted MVF so that computationally complex motion re-estimation and spatial-temporal pixel interpolation can be avoided.

In the simulation results, although we only show that the proposed MV processing method outperforms other conventional MV processing or motion estimation methods in MCFI, the possibilities of our work can be easily extended to other applications.

# Bibliography

[1] G. Dane and T. Q. Nguyen, “Smooth motion vector resampling for standard compatible video post-processing,” *Asilomar Conference on Signals, Systems and Computers*, vol. 2, pp. 1731–1735, Npv. 2004.

[2] S. Fujiwara and A. Taguchi, “Motion-compensated frame rate up-conversion based on block matching algorithm with multi-size blocks,” *International Symposium on Intelligent Signal Processing and Communication Systems, ISPACS 2005*, pp. 353–356, Dec. 2005.

[3] L. Alparone, M. Barni, F. Bartolini, and V. Cappellini, “Adaptively weighted vector-median filters for motion-fields smoothing,” *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 1996*, vol. 4, pp. 2267–2270, May 1996.

[4] A.-M. Huang and T. Nguyen, “Correlation-based motion vector processing with adaptive interpolation scheme for motion-compensated frame interpolation,” *IEEE Transactions on Image Processing*, vol. 18, pp. 740–752, April 2009.

[5] G. Haan, P. W. A. C. Biezen, H. Huijgen, and O. A. Ojo, “True-motion estimation with 3-D recursive search block matching,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, no. 5, pp. 368–379, Oct. 1993.

[6] R. Krishnamurthy, J. W. Woods, and P. Moulin, “Frame interpolation and bidirectional prediction of video using compactly encoded optical-flow fields and label fields,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 5, pp. 713–726, Aug. 1999.

[7] B.-D. Choi, J.-W. Han, C.-S. Kim, and S.-J. Ko, “Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 4, pp. 407–416, April 2007.

[8] T. Ha, S. Lee, and J. Kim, "Motion compensated frame interpolation by new block-based motion estimation algorithm," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 2, pp. 752–759, May 2004.

[9] G. D. Haan and P. W.A.A. Biezen, "An efficient true-motion estimation using candidate vectors from a parametric motion model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 1, pp. 85–91, Feb 1998.

[10] R. Braspenning and Gerard de H., "Efficient motion estimation with content-adaptive resolution," *Proceedings ISCE*, pp. E29–E34, Sept. 2002.

[11] R. Thoma and M. Bierling, "Motion compensating interpolation considering covered and uncovered background," *Signal Processing: Image Communication*, vol. 1, pp. 191–212, 1989.

[12] T. Chen, "Adaptive temporal interpolation using bidirectional motion estimation and compensation," *IEEE International Conference on Image Processin, ICIP 2002*, vol. 2, pp. 313–317, Sept. 2002.

[13] B.-D. Choi, S.-H. Lee, and S.-J. Ko, "New frame rate up-conversion using bi-directional motion estimation," *IEEE Transactions on Consumer Electronics*, vol. 46, pp. 603–609, Aug 2000.

[14] J. Zhai, J. Li K. Yu, and S. Li, "A low complexity motion compensated frame interpolation method," *IEEE International Symposium on Circuits and Systems, ISCAS 2005*, vol. 5, pp. 4927–4930, May 2005.

[15] C. Yang, P. Tao, and S. Yang, "An adaptive frame interpolation algorithm using statistic analysis of motions and residual energy," *IEEE Workshop on Multimedia Signal Processing, MMSP 2008*, pp. 978–240, 2008.

[16] H. Blume, G. Herczeg, O. Erdler, and T. G. Noll, "Object based refinement of motion vector field applying probabilistic homogenization rules," *IEEE Transactions on Consumer Electronics*, vol. 48, no. 3, pp. 694–701, Aug. 2002.

[17] S.-C.Han and J. W. Woods, "Frame-rate up-conversion using transmitted motion and segmentation fields for very low bit-rate video coding," *IEEE International Conference on Image Processing, ICIP 1997*, vol. 1, pp. 747–750, Oct 1997.

[18] B.-D. Choi, J.-W. Han, C.-S. Kim, and S.-J. Ko, "Frame rate up-conversion using perspective transform," *IEEE Transactions on Consumer Electronics*, vol. 52, pp. 975–982, Aug. 2006.

[19] G. A. Lunter, "Occlusion-insensitive motion estimation for segmentation," *Proceedings SPIE*, vol. 2671, pp. 573–584, 2002.

[20] P. Csillag and L. Boroczky, “Frame rate conversion based on acceleration and motion-based segmentation,” *Proceedings of the European Symposium on Advanced Imaging and Network Technologies, Conference on Digital Compression Technologies and Systems for Video Communications*, pp. 438–448, Oct 1996.

[21] J. Astola, P. Haavisto, and Y. Neuvo, “Vector median filters,” *IEEE Proceedings*, vol. 78, pp. 678–689, April 1990.

[22] G. Dane and T. Q. Nguyen, “Motion vector processing for frame rate up conversion,” *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2004*, vol. 3, pp. 309–312, May 2004.

[23] S. Sekiguchi, Y. Idehara, K. Sugimoto, and K. Asai, “A low-cost video frame-rate up conversion using compressed-domain information,” *IEEE International Conference on Image Processin, ICIP 2005*, vol. 2, pp. 974–977, Sept. 2005.

[24] J. Zhang, L. Sun, S. Yang, and Y. Zhong, “Position prediction motion-compensated interpolation for frame rate up-conversion using temporal modeling,” *IEEE International Conference on Image Processing, ICIP 2005*, vol. 1, pp. 53–56, Sept. 2005.

[25] H. Sasai, S. Kondo, and S. Kadono, “Frame-rate up-conversion using reliable analysis of transmitted motion information,” *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2004*, vol. 5, pp. 257–260, May 2004.

[26] J. Wang, N. Patel, and W. Grosky, “A fast block-based motion compensation video frame interpolation,” *Asilomar Conference on Signals, Systems and Computers*, vol. 2, pp. 1740–1743, Nov. 2004.

[27] M. E. Al-Mualla, “Motion field interpolation for frame rate conversion,” *IEEE International Symposium on Circuits and Systems, ISCAS 2003*, vol. 2, pp. 652–655, May 2003.

[28] S.-H. Lee, O. Kwon, and R.-H. Park, “Weighted-adaptive motion-compensated frame rate up-conversion,” *IEEE Transactions on Consumer Electronics*, vol. 49, no. 3, pp. 485–492, Aug. 2003.

[29] O. A. Ojo and G. Haan, “Robust motion-compensated video upconversion,” *IEEE Transactions on Consumer Electronics*, vol. 43, no. 4, pp. 1045–1056, Nov. 1997.

[30] Y.-T. Yang, Y.-S. Tung, and J.-L. Wu, “Quality enhancement of frame rate up-converted video by adaptive frame skip and reliable motion extraction,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 12, pp. 1700–1713, Dec. 2007.

[31] G. Dane, K. El-Maleh, and Y.-C Lee, “Encoder-assisted adaptive video frame interpolation,” *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2005*, vol. 2, pp. 349–352, March 2005.

[32] G. Dane and T. Q. Nguyen, “Optimal temporal interpolation filter for motion-compensated frame rate up conversion,” *IEEE Transactions on Image Processing*, vol. 15, pp. 978–991, Apr. 2006.

[33] A.-M. Huang and T. Nguyen, “Motion vector processing based on residual energy information for motion compensated frame interpolation,” *IEEE International Conference on Image Processing, ICIP 2006*, vol. 4, pp. 353–356, Sept. 2006.

[34] A.-M. Huang and T. Nguyen, “A novel motion compensated frame interpolation based on block merging and residual energy,” *IEEE Workshop on Multimedia Signal Processing, MMSP 2006*, vol. 4, pp. 353–356, Sept. 2006.

[35] A.-M. Huang and T. Nguyen, “A novel multi-stage motion vector processing method for motion compensated frame interpolation,” *IEEE International Conference on Image Processing, ICIP 2007*, pp. 389–392, 2007.

[36] A.-M. Huang and T. Nguyen, “A multistage motion vector processing method for motion-compensated frame interpolation,” *IEEE Transactions on Image Processing*, vol. 17, no. 5, pp. 694–708, May 2008.

[37] A.-M. Huang and T. Nguyen, “Motion vector processing using bidirectional frame difference in motion compensated frame interpolation,” *IEEE Workshop on Mobile Video Delivery*, pp. 1–6, June 2008.

[38] A.-M. Huang and T. Nguyen, “Motion vector processing using the color information,” *submitted to IEEE International Conference on Image Processing, ICIP 2009*, 2009.

[39] A.-M. Huang and T. Nguyen, “Correlation-based motion vector processing for motion compensated frame interpolation,” *IEEE International Conference on Image Processing, ICIP 2008*, pp. 1244–1247, Oct 2008.

[40] A.-M. Huang and T. Nguyen, “Motion vector processing based on trajectory curve analysis for motion compensated frame interpolation,” *submitted to IEEE International Conference on Image Processing, ICIP 2009*, 2009.

[41] A.-M. Huang and T. Nguyen, "True motion processing based on motion trajectory curve analysis and its application to motion compensated frame interpolation," *submitted to IEEE Transactions on Image Processing*, 2009.

[42] Kaup A and T. Aach, "Efficient prediction of uncovered background in interframe coding using spatial extrapolation," *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 1994*, vol. 5, pp. 501–504, 1994.

[43] T. Gevers and H. Stokman, "Classifying color edges in video into shadow-geometry, highlight, or material transitions," *IEEE transactions on multimedia*, vol. 5, no. 2, pp. 237–243, June 2003.

[44] P. E. Eren, Y. Altunbasak, and A. M. Tekalp, "Region-based affine motion segmentation using color information," *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 1997*, pp. 3005–3008, 1997.

[45] N.S. Lee, T. Yamasaki, and K. Aizawa, "Hierarchical mesh decomposition and motion tracking for time-varing-meshes," *IEEE International Conference on Multimedia and Expo, ICME 2008*, pp. 1565–1568, 2008.

[46] T. D. Nguyen, "A novel motion compensated frame interpolation method for improving side information in distributed video coding," *Proceedings International Symposium on Information Technology Convergence*, pp. 179–183, Nov. 2007.

[47] "Information technology – coding of audio-visual objects – part 2: Visual," *ISO/IEC DIS 14496-2.*, 2001.

[48] "Video coding for low bitrate communication," *International Telecommunications Union, ITU-T Recommendation H.263*, 1998.

[49] Z. Wang, A. C. Conrad, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, April 2004.

[50] "Advanced video coding for generic audiovisual services," *International Telecommunications Union, ITU-T Recommendation H.264*, 2003.