TRUE MOTION ESTIMATION FOR VIDEO FRAME RATE UP-CONVERSION

BURAK ÇİZMECİ
B.S., Electronics Engineering, Işık University, 2007
B.S., Computer Engineering, Işık University, 2008

Submitted to the Graduate School of Science and Engineering
in partial fulfillment of the requirements for the degree of
Master of Science
in
Electronics Engineering

IŞIK UNIVERSITY
2009

IŞIK UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

TRUE MOTION ESTIMATION FOR VIDEO FRAME RATE UP-CONVERSION

BURAK ÇİZMECİ

APPROVED BY:

Assist. Prof. Hasan F. Ateş (Thesis Supervisor)	Işık University	____________________
Assist. Prof. Onur Kaya	Işık University	____________________
Assist. Prof. M. Taner Eskil	Işık University	____________________

APPROVAL DATE: ..../..../....

TRUE MOTION ESTIMATION FOR VIDEO FRAME RATE UP-CONVERSION

Abstract

Since the invention of high definition display technologies, video standards conversion problem has become an important task to store, transmit and display new video formats. One of the standards conversion issue, video frame rate up-conversion (FRUC) should be considered as an important problem for high definition displays because these displays reach high refresh rates up to 100 Hz and low video frame rates should be increased before displaying. The solutions generated by conventional FRUC techniques cause artifacts such as jerky motion and judder effect on the screen. The objective of this thesis is to remove judder and occlusion artifacts and generate a smooth object motion for high definition displays. In order to solve the problem, this thesis describes a multi-stage motion vector (MV) post-processing technique for true motion estimation and proposes a new video frame rate up-conversion method with occlusion adaptive overlapped block motion compensation, which increases the temporal resolution without loss of spatial resolution and consistency. Conventional block matching algorithms yields us unreliable MVs which should be post-processed in the refinement stages to get a consistent MV field in spatial domain. Relations between neighboring MVs are utilized to detect occlusion regions. Overlapped Block Motion Compensation (OBMC) is adapted to these problematic regions in order to reduce halo artifacts caused by occlusion. Compared to existing methods, the proposed algorithm achieves FRUC with reduced loss of spatial resolution and reduced amount of artifacts.

VİDEO ÇERÇEVE HIZ ARTIRIMI İÇİN GERÇEK DEVİNİM VEKTÖR KESTİRİMİ

Özet

Yüksek çözünürlüklü görüntüleme teknolojilerinin icadı ile yüksek çözünürlük videoyu saklma, iletme ve görüntüleme formatlarındaki değişiklikler video standartları çevirimi problemini önemli bir konuma getirmiştir. Standart çevirim problemi olan video çerçeve hız artırımı yüksek çözünürlüklü ekranlar için önemli bir sorun olarak kabul edilmelidir çünkü bu ekranların yenileme hızları 100 Hz'e kadar çıkabilmektedir ve düşük çerçeve hızına sahip bir videonun bu ekranlarda görüntülenmeden önce ön işlemlerden geçirilip çerçeve hız artırımı sağlanmalıdır. Klasik yöntemlerle sağlanan çerçeve hız artırımı ekranda düzensiz hareketlere ve titreme etkisine neden olmaktadır. Bu tezin amacı video çerçeve hız artırımında titreme etkisini ve kapanma /açılma yapaylıklarını kaldırmak ve çerçeveler arasında yumuşak bir nesne hareketi oluşturmaktır. Bu tez çalışması sorunu çözmek amacıyla gerçek devinim vektör alanı üreten çok aşamalı devinim vektör art işlemlerini tanımlamakta ve çerçeve hız artırımı için kapanma/açılma uyarlamalı örtüşmeli blok devinim denkleştirmesi yöntemini önermektedir. Klasik devinim kestirim (DK) algoritmalarının verdiği düşük güvenirlikli devinim vektörleri (DV) çok aşamalı iyileştirme art-işlemlerine tabi tutulur. Uzamsal olarak tutarlı bir DV alanı elde edildikten sonra komşu vektörler arasındaki ilişkiler kullanılarak kapanma/açılma bölgeleri tespit edilir. Örtüşmeli blok devinim denkleştirme filtresi (ÖBDD) tespit edilen bu bölgelere uyarlanılarak kapanma/açılmadan kaynaklanan yapaylıkların azaltılması sağlanır. Böylece varolan ÇHA yöntemlerine göre uzamsal çözünürlükten daha az ödün verilerek ve daha az yapaylıkla ÇHA gerçekleştirebilen bir sistem önermekteyiz.

Acknowledgements

There are many people who helped to make my years at the graduate school most valuable. First, I thank my supervisor, Assist. Prof. Hasan Fehmi Ateş. Having the opportunity to work with him over the years was intellectually rewarding and fulfilling. I also thank Assoc. Prof. Uluğ Bayazıt who gives me the motivation of this research starting from the early stages of my academic career. I thank him for his insightful suggestions and expertise. Many thanks to VESTEK Research and Development company digital video processing algorithm design department, who answered my questions and problems on video processing. I also thank my research colleague, Res. Assist. Engin Tamer, who never leaved me alone to generate solutions to our technical problems and I was glad to work with him. I would also like to thank to my graduate student colleagues, Res. Assist. Murat İşleyen and Res. Assist. Mehmet Güneş who helped me all through the years full of class work and exams. The last words of thanks go to my family. I thank my parents Hüseyin Cahit, Hatice Demet, my brother Kerem and my sister Nisa for their patience and encouragement. Lastly, I thank my house mates, Reha Yavuz and Yusuf İzzettin Horasanlı, for their endless support through this long journey.

This study was supported by The Scientific and Technological Research Council of Turkey (TÜBİTAK) Grant No: 108E201

To my family…

Table of Contents

Abstract ii

Özet iii

Acknowledgements iv

List of Tables viii

List of Figures ix

List of Symbols x

List of Abbreviations xi

1 Introduction 1
- 1.1 Video Picture Rate Up-Conversion 2

2 Motion Estimation for Video Frame Rate Up-Conversion 5
- 2.1 Motion Estimation Methods 5
- 2.2 Block-Based Motion Estimation 7
  - 2.2.1 Three Step Search Block Matching Algorithm 9
  - 2.2.2 Hexagonal Search Block Matching Algorithm 10
- 2.3 True Motion Estimation 12
  - 2.3.1 3-D Recursive Search Block Matching Algorithm 13
  - 2.3.2 A Multi-Stage Motion Vector Post-Processing for True Motion Estimation 15
    - 2.3.2.1 Motion Vector Reliability Classification 15
    - 2.3.2.2 Macroblock Merging Based on Motion Vector Reliability 16
    - 2.3.2.3 Motion Vector Selection 18
    - 2.3.2.4 Motion Vector Re-classification Based on Bi-directional Prediction Difference 19
    - 2.3.2.5 Motion Vector Refinement 19

3 Occlusion Aware Motion Compensation Scheme for Video Frame Rate Up-Conversion 21

3.1 Handling Occlusion Problem 21
3.1.1 Occlusion Detection with Double ME 23
3.1.2 Occlusion Detection Based on Multiple Frame Analysis . . 24
3.1.3 Occlusion Detection From Vector Field 25
3.2 Gradient Based Motion Vector Smoothing 26
3.3 Occlusion Detection for Motion Compensated Frame Rate Up-Conversion 28
3.4 An Occlusion Adaptive Overlapped Block Motion Compensation . 30

4 Simulation Results and Comparisons 33
4.1 Objective Quality Measures for Video Processing 34
4.1.1 Peak Signal To Noise Ratio 34
4.1.2 Structural Similarity Test 34
4.2 Subjective Quality Measure for Video Processing 36
4.2.1 Subjective Mean Squared Error 36
4.3 Simulation Results 38

Conclusion 50

References 50

Curriculum Vitae 55

List of Tables

2.1 PSNR comparison on Football sequence [11] 11

4.1 PSNR Comparison with [10] 42
4.2 HS and 3DRS performance comparison 43
4.3 PSNR and SSIM measurements 44
4.4 MSE and SMSE measurements 45
4.5 Performance comparison with Algorithm 1 46
4.6 Performance comparison with journal [5] 46

List of Figures

2.1 Block-based motion estimation 8
2.2 Three Step Search Block Matching 9
2.3 Hexagonal Search Block Matching 10
2.4 The demonstration of spatial and temporal prediction vectors . . 13
2.5 3-D Recursive Search Block Matching Algorithm 14
2.6 MB Merging Shapes 16
2.7 MB Merging Examples 17
2.8 Bi-directional Search demonstration 18

3.1 Halo artifact caused by occlusion problem 22
3.2 Four field occlusion analysis 24
3.3 Occlusion detection from vector field 25
3.4 Gradient based search for MV smoothing 26
3.5 Covered and uncovered handling in MC-FRUC 28
3.6 The demonstration of adjacent MBs for Occlusion Detection . . . 29
3.7 The demonstration of adjacent MBs for Overlapped-Block Motion Compensation 30
3.8 The Overall Frame Rate Up-Conversion System 32

4.1 Tested video frames 38
4.2 Adaptive OBMC [6] 39
4.3 The proposed Occlusion aware OBMC scheme 39
4.4 Adaptive OBMC [6] 40
4.5 The proposed Occlusion aware OBMC scheme 40
4.6 Adaptive Frame Rate Conversion [7] 41
4.7 The proposed Occlusion aware OBMC scheme 41
4.8 Test sequences on table 4.1 42
4.9 Table Tennis video sequence results 47
4.10 Adjacent test frames in Girl sequence 48
4.11 OBMC scheme without Gradient ABPD smoothing 49
4.12 OBMC scheme with Gradient ABPD smoothing 49

List of Symbols

v	Vector
f	Video frame
$\varepsilon(\overrightarrow{v})$	Error function
N	Block size
B	Motion block
i, j	Pixel index
$\overrightarrow{U}_k$	Update vector
$E_{m,n}$	Error energy
Y	Luminance component
Cb	Blue difference chroma component
Cr	Red difference chroma component
L_i	Reliability label
ε_i	Thresholds for error energy
$d_{i,j}$	Angular distance measure $0 \leqslant d_{i,j} \leqslant 2$
θ	Angle between two vectors
μ	Mean value of a block
σ	Standard deviation of a block

List of Abbreviations

ÇHA	**Ç**erçeve **H**ız **A**rtırımı
PC	**P**ersonal **C**omputer
CIF	**C**ommon **I**ntermediate **F**ormat
QCIF	**Q**uarter **C**ommon **I**ntermediate **F**ormat
SDTV	**S**tandard **D**efinition **T**ele**V**ision
HDTV	**H**igh **D**efinition **T**ele**V**ision
BBC	**B**rithish **B**roadcasting **C**orporation
DSP	**D**igital **S**ignal **P**rocessor
FPGA	**F**ield **P**rogrammable **G**ate **A**rray
ASIC	**A**pplication **S**pecific **I**ntegrated **C**ircuit
VLSI	**V**ery **L**arge **S**cale **I**ntegration
MB	**M**acro **B**lock
ME	**M**otion **E**stimation
MC	**M**otion **C**ompensation
MV	**M**otion **V**ector
BMA	**B**lock **M**atching **A**lgorithm
SAD	**S**um of **A**bsolute **D**ifferences
MSE	**M**ean **S**quared **E**rror
SMSE	**S**ubjective **M**ean **S**quared **E**rror
MPEG	**M**otion **P**ictures **E**xperts **G**roup
H.260	**H.260** familiy motion compensation based codec standards
MPC	**M**aximum **M**atching **P**el **C**ount
PSNR	**P**eak **S**ignal to **N**oise **R**atio
SSIM	**S**tructural **S**imilarity **T**est

HS	**H**exagonal **S**earch block matching algorithm
4SS	**F**our **S**tep **S**earch block matching algorithm
DS	**D**iamond **S**earch block matching algorithm
NTSS	**N**ew **T**hree **S**tep **S**earch block matching algorithm
2-D	**2 D**imension
3-D	**3 D**imension
3DRS	**3-D R**ecursive **S**earch block matching algorithm
Bi-LMC	**B**idirectional **L**inear **M**otion **C**ompensation
OBMC	**O**verlapped **B**lock **M**otion **C**ompensation
AOBMC	**A**daptive **O**verlapped **B**lock **M**otion **C**ompensation
FRUC	**F**rame **R**ate **U**p-**C**onversion
MC-FRUC	**M**otion **C**ompensated **F**rame **R**ate **U**p-**C**onversion
MVRM	**M**otion **V**ector **R**eliability **M**ap
BPD	**B**idirectional **P**rediction **D**ifference
ABPD	**A**bsolute **S**um of **B**idirectional **P**rediction **D**ifference
GradABPD	**Grad**ient **B**ased **A**bsolute **S**um of **B**idirectional **P**rediction **D**ifference
CS	**C**andidate **S**et of motion vectors
US	**U**pdate **S**et of motion vectors
RelPerf	**Rel**ative **Perf**ormance
med	**med**ian
Mod	**Mod**ulo operation
Thr	**Thr**eshold

Chapter 1

Introduction

Digital video standards are very important for many digital video systems which have different spatial and temporal resolution requirements. Moreover, different format standards for storage, transmission and display technology of the digital video signal come to the fore as tasks that have to be solved. The conversion of digital video from one standard to another is a challenging problem that has to be done without losing the signal quality. The standards conversion consists of frame/field rate down/up-conversion, interlacing/de-interlacing and spatial interpolation/scaling. Frame rate up-conversion refers to the increase in temporal sampling rate of a progressive or interlaced video. De-interlacing operation converts interlaced video into progressive video. Spatial interpolation/scaling is needed when the number of pixels in a video line is different from the number of pixels in a line on the display. Therefore, standards conversion techniques allow us to exchange video signal information between different display technologies [1].

In recent years, we are faced with new inventions such as video conferencing, HDTV, workstations and PCs which lead to different video formats. Video formats such as CIF and QCIF have smaller picture size and lower frame rates for video communication, progressive and interlaced HDTV formats support from 50 and 60 Hz and other video formats used on computer workstations, PC monitors and enhanced television displays support field rates up to 100 Hz [2]. Especially, enhanced HDTV displays bring us new visual tastes because we get used to watch high quality video that are recorded in HDTV format. On the other hand, when a low quality video signal is displayed on this screen, the video processor applies standards conversion techniques on the low quality video. As a result, the output video signal is not as satisfactory as high quality HDTV format and has artifacts caused by conversion, this problem leads to reconsideration of the standards conversion. Therefore, the research of the standards conversion algorithms have become more popular than in the past.

In this dissertation, we focus on video frame rate up-conversion problem which is the basic standards conversion task described above. For example, a digitally recorded video with a temporal sampling rate 25 frames/sec is desired to be converted into a video format at 50 frames/sec. In an other point of view, video frame rate up-conversion is not only a standards conversion but also a video enhancement tool which increases the temporal resolution of the digital video aiming to improve the quality of high definition displays. In the following section, the requirements to solve video frame rate up-conversion problem is described and a brief summary of the literature is presented to give information about the recent research and achievements.

1.1 Video Picture Rate Up-Conversion

Enhancing the temporal resolution of a video coded at a low temporal resolution has been a popular research problem especially in consumer electronic industries. In the past, non-motion compensated adaptive filtering methods were popular. These methods are not good enough in generating high quality output video signal and mostly cause annoying artifacts such as motion blur and motion judder caused by frame repetition. Recent developments in hardware technology lead to a rise in processing power of DSPs, FPGAs and ASICs. High computational power enables us to design high complexity algorithms and consequently, the motion compensated (MC) video frame rate up-conversion (FRUC) schemes become very popular. The motion estimation (ME) stage is the most important part of the MC-FRUC system due to the fact that a true motion vector (MV) representation of moving objects is essential for the interpolation stage. Unfortunately, block matching algorithms (BMA) that are commonly used fail in some problematic cases such as covering/uncovering regions and blocks containing multiple moving objects. To utilize BMA efficiently, we should refine the wrong MVs in the output MV field and detect the occluded areas between the adjacent fields because covered/uncovered parts of the video frame cause occlusion artifacts which are hollow, ghost and blocking artifacts at the moving object boundaries.

In the literature, two classes of true ME algorithms come to the fore. The first class was proposed by de Haan *et al.*[3], 3-D Recursive Search Block Matching which contains smoothness constraints integrated to the estimator. The second class consists of algorithm types which apply MV post-processing techniques on the output MV field of BMA to provide temporal and spatial smoothness consistency. In [4], Nguyen *et al.* proposed a multi-stage MV post-processing method for video frame rate up-conversion. At the interpolation stage in [4] bidirectional linear motion compensation is applied. In [5], Nguyen made some enhancements over the previous system. He changed the multi-stage MV post-processing method by adding a neighborhood consistency constraint and additional to this, a bidirectional MC which is adapted to the occlusion areas is applied. In [6], Jea-Ko *et al.* proposed an adaptive overlapped block motion compensation (AOBMC) for video frame rate up-conversion which uses a sigmoid window function for weight calculation of overlapped blocks. In [7], Gao *et al.* proposed an adaptive frame rate up-conversion based on motion classifications. Gao *et al.* uses a new error metric, correctional sum of absolute differences (CSAD), to eliminate motion errors that occurred due to the errors caused by SAD and the system adapts itself to the sensitive regions: occlusion, background and foreground at the interpolation stage.

In this thesis, we consider both classes of true ME algorithms in [4] and [3] in chapter 2. In chapter 3, contrasting with MC-FRUC systems in [4], [6] and [7], we propose an occlusion artifact removing MC scheme using an adaptive OBMC at the interpolation stage. Before the interpolation stage, we correct and smooth out the output MV field using a gradient based technique which is sensitive to object boundaries and then apply occlusion detection to weed out the problematic regions which are covered or uncovered on the frame being interpolated. At the interpolation stage, occlusion adaptive OBMC smooths out the blockiness along the object boundaries and a smooth picture is interpolated between the adjacent video frames. In conclusion, occlusion adaptive OBMC provides better visual quality for human vision system which is very sensitive to the artifacts occurring on the object boundaries.

At the beginning of chapter 3, we focus on the occlusion problem because the most important issue in this research is the occlusion problem at the interpolation stage. In [2], de Haan studied the problem deeply and compared MC-FRUC systems that are using the occlusion data. According to the subjective quality comparisons in [2], a four field interpolation scheme which is proposed by BBC research in [8] has the best visual quality among other interpolation methods. However, MC-FRUC system proposed in [8] is a very complex algorithm and it is difficult to utilize the algorithm in today's consumer products such as HDTVs and hand held mobile devices. In this research, we aimed to design an interpolation scheme which is not too complex and provide high quality interpolated pictures for the human vision system.

In chapter 4, the objective quality measurements, PSNR (Peak signal to noise ratio), SSIM (Structural similarity test), MSE (Mean squared error) and SMSE (Subjective mean squared error)are presented in detail and the reliability of these measurements are discussed according to the subjective quality of the interpolated pictures. The simulation results of the proposed method is compared subjectively and objectively with the methods in [7], [6], [9], [5] and [10].

Chapter 2

Motion Estimation for Video Frame Rate Up-Conversion

Digital video processing literature contains many research topics and most of them need a motion estimation stage to reveal the correlations between adjacent frames. According to this reason, motion estimation takes an important role in most applications. If motion estimation fails, the designed algorithm could even have worse performance than the case when no motion information is used. Based on these considerations, motion estimation has a top-line position in video processing research. In this chapter, the motion estimation methods are introduced and the most important ones, block matching algorithms, are discussed in detail. Finally, the difference between conventional motion estimation and true motion estimation is considered and two important true motion estimation systems are discussed in detail.

2.1 Motion Estimation Methods

In video processing, motion estimation deals with the motion of moving objects which are projected on a 2-D image plane or alternatively, 2-D position change of objects on the image plane. According to the type of the application, motion estimation algorithms have varying approaches with specific focus on the application.

The most commonly used methods can be stated as follows:

- Optical-flow methods
- Pel-recursive methods
- Block matching
    - Hierarchical block matching

Motion is usually defined in terms of a motion or a displacement vector like in equation 2.1:

$$\vec{v} = (v_1, v_2)^T \tag{2.1}$$

All four motion estimation methods mentioned above are based on the following constraint equation 2.2.

$$f(i, j, t) = f(i - v_1, j - v_2, t - 1) \tag{2.2}$$

In equation 2.2, illuminance change is ignored and assumed to be constant over time and the main idea is to find the best match of pixel (i,j,t) in the previous picture (t-1). This basic equation can be solved approximately in the least-squares sense. Therefore, an error function should be minimized and the best matching position is chosen accordingly. Analytically, the general minimization can be stated as in equation 2.3

$$\vec{v}^* = \arg\min_{v} \varepsilon(\vec{v}) \tag{2.3}$$

This kind of matching problem can be defined simply as the correlation of neighboring information. But the problem is complicated because during projection on 2-D plane, the information on the third coordinate of the 3-D world is lost. As a result it becomes difficult to estimate especially the motions toward or away from the camera, rotation and 3-D deformable object motion. Therefore, the motion estimation defined above estimation is referred as 2-D motion estimation. Additional to these problematic object motions, the covering/uncovering problem is an other issue that challenges motion estimation algorithms. In the case of covering/uncovering problem, the information (pixels) that is searched in the reference picture is not present in the search area, therefore the estimation error has a high value on these problematic regions.

2.2 Block-Based Motion Estimation

The most convenient and popular method, block-based motion estimation is widely used in VLSI implementations, H.260 and MPEG family codecs. Block matching ME algorithms find the best motion vector by using a spatial domain search in a limited area. In figure 2.1, the dashed window demonstrates the search area of the block matching and the block centered in the current frame is searched in the previous frame. The displacement among the blocks is measured in pixel domain by simply calculating the analytic difference between pixel positions. The basic properties of block matching algorithms is as follows:

- The matching criteria (e.g. maximum cross correlation, minimum error).
- The search strategy (e.g. three step search, diamond search).
- The determination of block size (e.g. adaptive, hierarchical, fixed).
- The size of the search window (changes according to block size).

The matching criteria is the most important part of the block matching algorithm. Since the ME problem can be solved in least squares sense, the mean squared error (MSE) is a good choice for the error criteria. In equations 2.4 and 2.5, the MSE is defined for ME problem and is used as a constraint function. The best vector is chosen according to this minimization.

$$MSE(v_1, v_2) = \frac{1}{N \times N} \sum_{(i,j) \epsilon B} [f(i,j,t) - f(i - v_1, j - v_2, t-1)]^2 \qquad (2.4)$$

$$[\vec{v_1}, \vec{v_2}]^T = \arg \min_{(\vec{v_1}, \vec{v_2})} MSE(v_1, v_2) \qquad (2.5)$$

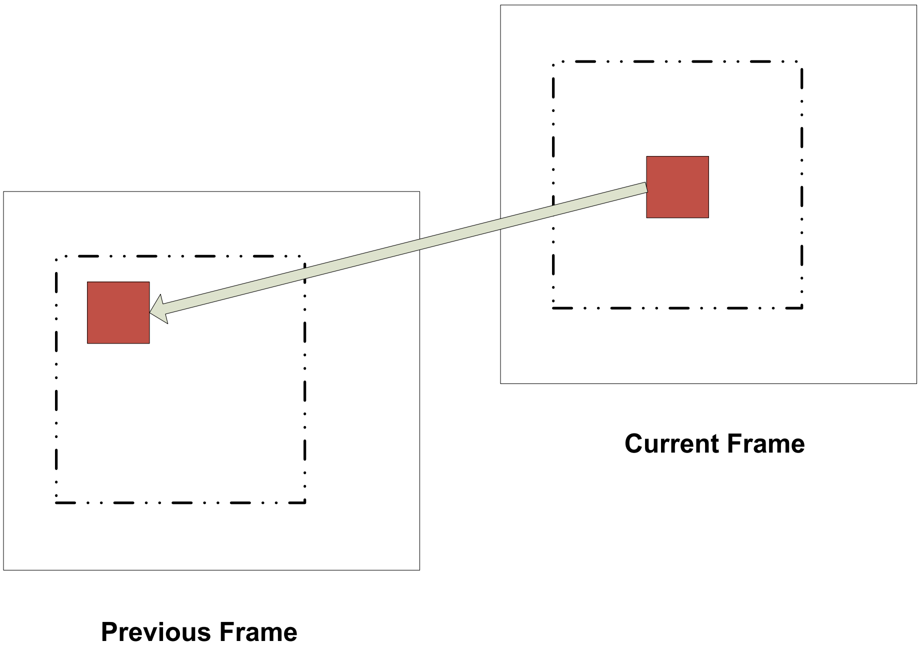


Figure 2.1: Block-based motion estimation

Instead of using MSE criterion, sum of absolute differences (SAD) is a better choice for real-time processing environments because the computation cost of squared operator is very high in VLSI implementations. We can state SAD criterion and its minimization operation in equations 2.6, 2.7:

$$SAD(v_1, v_2) = \frac{1}{N \times N} \sum_{(i,j)\epsilon B} |f(i, j, t) - f(i - v_1, j - v_2, t - 1)| \tag{2.6}$$

$$[\vec{v_1}, \vec{v_2}]^T = \arg \min_{(\vec{v_1}, \vec{v_2})} SAD(v_1, v_2) \tag{2.7}$$

Furthermore, there are other matching criteria such as maximum matching pel count (MPC)[1] but SAD is the most popular and the easiest way to reach the minimum error. In this chapter, we discuss ME algorithms that are operating in pixel domain [1]. There are also other methods that works in frequency domain too. The phase correlation function is used to find the motion vectors. However, frequency domain approaches are not so popular due to its computation costs.

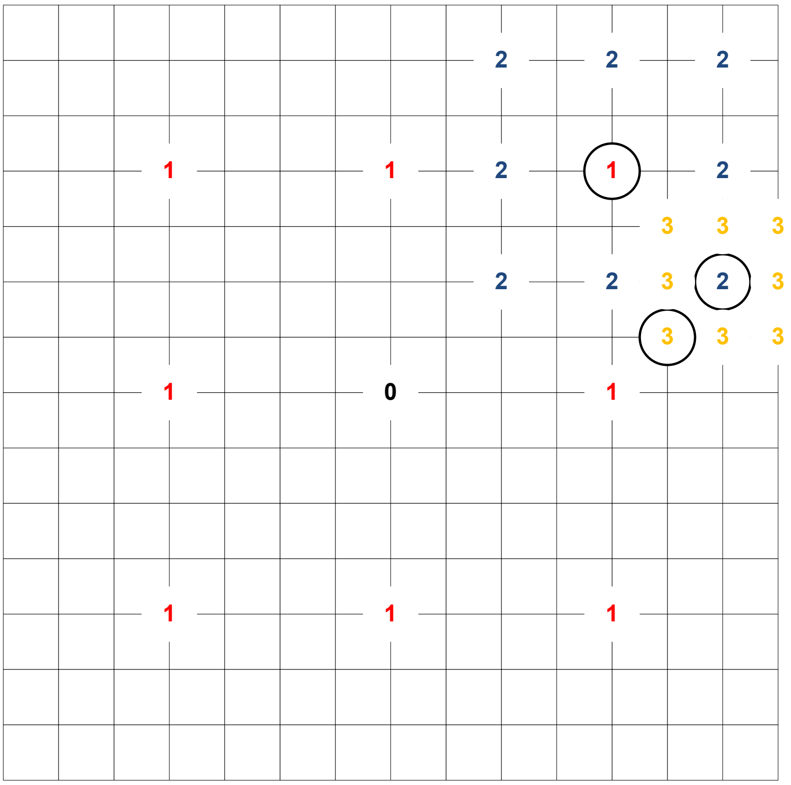


Figure 2.2: Three Step Search Block Matching

2.2.1 Three Step Search Block Matching Algorithm

Mainly motion estimation algorithms differ in search strategy. Three step search block matching is a greedy approach towards motion estimation. In three steps, the algorithm reaches a result but it does not guarantee that the found solution is optimum. However, it is very simple and time saving method compared to other strategies. The algorithm is illustrated in figure 2.2 and it is composed of three steps. Firstly, it begins calculating SAD at the positions 0 and 1s then the search center moves to best matching position and the same procedure is repeated for the positions 2s but this time search range is decreased. After finding a better match among 2s, the search center moves to the best match and the same procedure is done thirdly for the positions 3s. Finally, we found a minimum match error position in a limited search area. The resulting motion vector is calculated by taking the vector difference between the initial position and the final position.

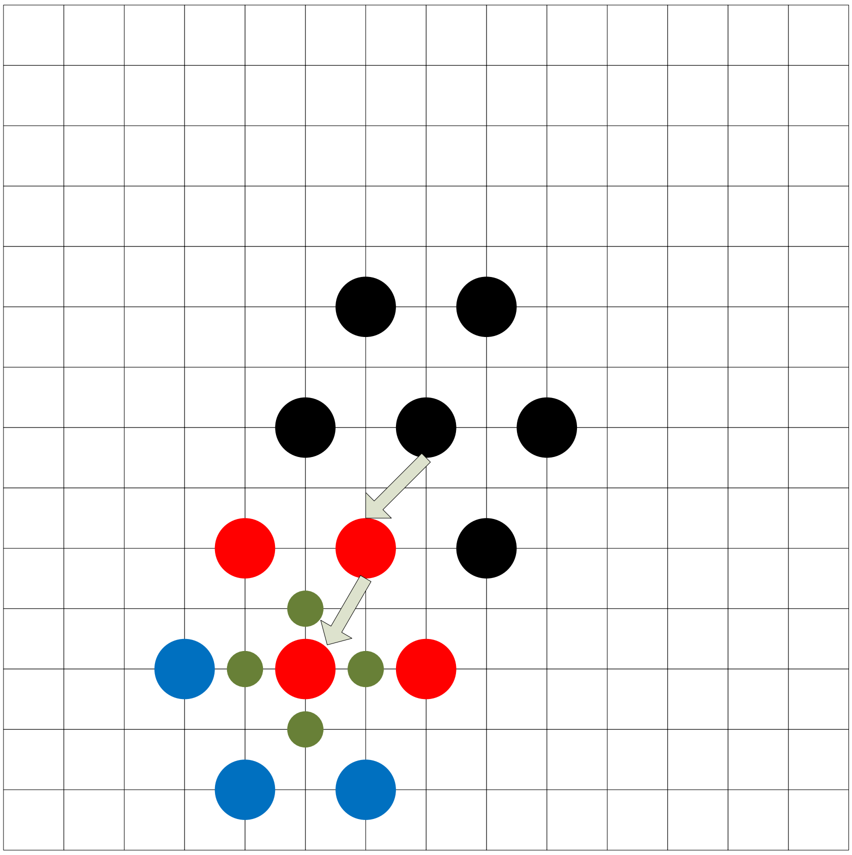


Figure 2.3: Hexagonal Search Block Matching

2.2.2 Hexagonal Search Block Matching Algorithm

As seen on figure 2.3, hexagonal search algorithm has a central biased search pattern with hexagonal shaped 7 check points (at the edges). The algorithm is composed of three steps. As an example, in figure 2.3 the search begins at the center and check for minimum SAD value among 7 points around the hexagon shape. In this scenario, the point on the below left gives the minimum SAD and then the search center moves to that point and a new hexagon is created around this point. The same procedure is repeated on the second hexagon and the center moves to below right point on the second hexagon. Finally, on the third hexagon we got the minimum SAD value at the center of the hexagon. The search is going on the first order neighbors of the minimum SAD point and the best position is chosen according to minimum SAD value.

After this example, we can state the algorithm steps as follows:

- **Step 1:** The initial hexagon is centered at the middle and SAD calculation is done at 7 check points. If minimum SAD is found at the center of the hexagon, then go to Step 3, otherwise go to Step 2.

- **Step 2:** If the minimum SAD is found among 6 points around the hexagon, then the search center moves to that point. SAD calculation is done on the new points around newly generated hexagon. If minimum SAD is found at the center, then go to Step 3, otherwise recursively repeat Step 2 until reaching Step 3.

- **Step 3:** This time we change the search pattern from hexagon to star. SAD is calculated among first order touching neighbors of the hexagon center. The point that gives minimum SAD is chosen for the best match and the motion vector is calculated by simple vector difference.

In our algorithms we used hexagonal search block matching (HS) as an initial motion field estimator. The reason that we chose hexagonal search block matching as an initial estimator is its low complexity and high PSNR results among other motion estimators which are 4 step search (4SS), new three step search (NTSS) and diamond search (DS) [11]. On table 2.1, we can see that HS outperforms other block matching strategies when it is tested on Football video sequence. HS is good choice for predictive video coding. However, we should post-process output motion vectors of HS in order to reach a true motion vector field.

Table 2.1: PSNR comparison on Football sequence [11]

Algorithms	**Complexity**	**Average PSNR (dB)**
FS	100 %	30.56
4SS	10.18 %	30.54
NTSS	10.34 %	30.54
DS	9.09 %	30.55
HS	7.91 %	30.55

2.3 True Motion Estimation

For video enhancement applications, motion estimation should represent true motion information of objects. Because of that block based motion estimation is not adequate to represent the true motion by itself. In the field of true motion estimation research, we are faced with two types of true motion trackers. One of them is 3-D recursive search block matching which is designed by de Haan *et al.*[3]. Second method is MV post-processing which performs post-processing on the motion vector field after getting an initial motion field from a conventional motion estimator such as hexagonal search block matching. In the following sections, we introduce 3-D recursive search block matching [3] and a multi-stage MV post-processing algorithm [4] for video frame rate up-conversion systems. In order to get a reliable true motion field, the output motion vectors of block based motion estimator should be post-processed before motion compensation (MC) like in the multi-stage method [4] or the initial estimator should be supported by temporal and spatial constraints like in the 3-D recursive search method [3].

To achieve a true MV field, an estimator should provide the following constraints [12]:

- **Block Size:** Begin with large blocks (16×16) and then refine the block size up to (4×4) to provide smoothness on the object boundaries.
- **Spatial Consistency:** The motion vectors of spatially neighboring blocks should be consistent and smooth.
- **Temporal Consistency:** The motion vectors of temporally neighboring blocks should be consistent and smooth.
- **Tracking Problematic Regions:** Occlusion regions and object boundaries should be tracked to provide a correct estimation because these regions are difficult to estimate due to covering/uncovering.

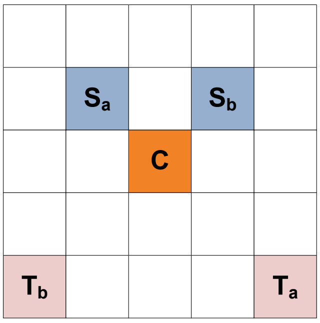


Figure 2.4: The demonstration of spatial and temporal prediction vectors

2.3.1 3-D Recursive Search Block Matching Algorithm

Unlike conventional block matching methods, 3-D recursive search block matching (3DRS) does not have a well-defined search range that is used for testing candidate MVs. Cleverly, 3DRS algorithm forms its possible predictions from its spatiotemporal neighbors. Thus this approach reduces the risk of converging a wrong minimum point. 3DRS is based on two assumptions:

- Objects are assumed to be larger than motion blocks.
- Objects have inertia.

First assumption gives us the clue that we can find the velocity of the current block in at least one of the neighboring blocks. However in this case we are faced with the causality problem, not for every neighboring block a MV has been estimated yet. To overcome this problem de Haan [2] suggests to utilize the corresponding neighbor blocks in the temporal domain (in the previous motion vector field). This suggestion is generated from the second assumption above which implies that physically a moving object would like to preserve its state of motion. The positions of the spatial and temporal candidate MVs are shown in figure 2.4 and additional candidate MVs are generated from the spatial neighbors by adding a random update vector. The analytical form of the candidate MVs is given in equation 2.8.

$$
CS(\underline{B},t) = \begin{cases} \overrightarrow{v}(i-1,j-1,t), \overrightarrow{v}(i-1,j+1,t) \\ \overrightarrow{v}(i-2,j-2,t-1), \overrightarrow{v}(i-2,j+2,t-1) \\ \overrightarrow{v}(i-1,j-1,t) + \overrightarrow{U}_k, \overrightarrow{v}(i-1,j+1,t) + \overrightarrow{U}_k \end{cases} \tag{2.8}
$$

Where $\overrightarrow{U}_k \epsilon US$ demonstrates the random update vector which is taken from a fixed set randomly. The update set can be as follows:

$$
US = \begin{cases} \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 0 \\ -2 \end{pmatrix} \\ \\ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 3 \\ 0 \end{pmatrix}, \begin{pmatrix} -3 \\ 0 \end{pmatrix} \end{cases} \tag{2.9}
$$

Finally, the best motion vector is chosen from the candidate set as the one that minimizes SAD error function which is described in the previous sections. Additionally, the candidate set can be extended by adding global motion models which is derived from the previous motion field. This new information, global motion vectors, provides an advantage that will increase the accuracy of the generated candidate set in time. In figure 2.5 a general system overview is given with the global motion models.

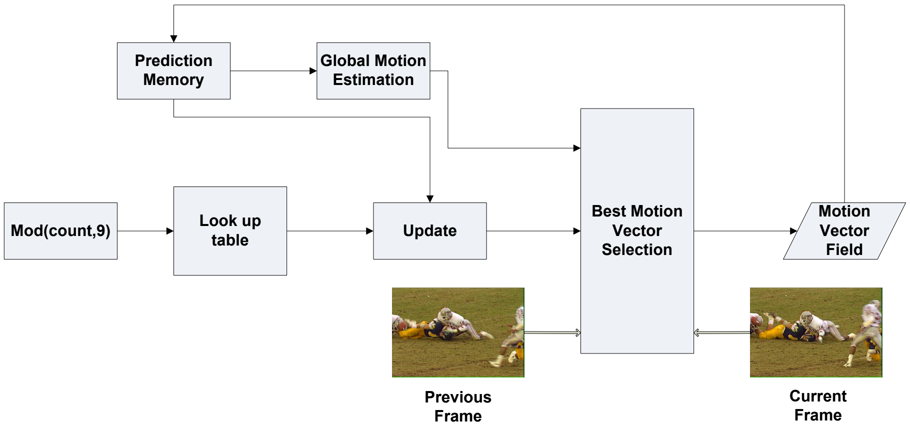


Figure 2.5: 3-D Recursive Search Block Matching Algorithm

2.3.2 A Multi-Stage Motion Vector Post - Processing for True Motion Estimation

When the visual quality of block based ME algorithms is considered, they have significant problems on the edge discontinuities and corrupt the structure of the objects. Block based ME algorithms assign only one motion vector for an 8 by 8 pixel block. Therefore, the block based ME algorithms in the literature can not deal with the problem when there is more than one motion in an 8 by 8 pixel block. Purely applying Block based ME and MC to FRUC problem yields annoying blocking artifacts and deformations on object boundaries. Human eye is very sensitive to these high frequency changes on scenes. On the other hand, low frequency changes at object boundaries such as blurring can not be easily detected by human eye. A true motion field should be generated for objects and a consistent motion field should be provided when compared to their previous and next positions. The method considers the reliability of each motion vector that is received from a Block Matching ME algorithm. By analyzing the distribution of residual energies and effectively merging the blocks according to unreliable connectivity between neighboring blocks, a hierarchical refining of the motion vectors which are in the unreliable set is desired.

2.3.2.1 Motion Vector Reliability Classification

The first stage of the motion vector processing algorithm divides the motion vector space into reliability classes. In the following stages the algorithm focuses on the unreliable motion vectors and corrects them. Minimum SAD criterion is calculated only over the luminance domain. Consequently, this approach may often result in mismatches on the color space. Therefore, minimizing SAD in luminance domain is inadequate to represent the actual prediction error. Motion vector reliability classification stage considers this color mismatch problem by using both luminance and chrominance information in residual energy calculation in equation 2.10.

$$E_{m,n} = \sum_{(i,j)\epsilon b^{Y}_{m,n}} |r_Y(i,j)| + \alpha(\sum_{(i,j)\epsilon b^{Cb}_{m,n}} |r_{Cb}(i,j)| + \sum_{(i,j)\epsilon b^{Cr}_{m,n}} |r_{Cr}(i,j)|) \qquad (2.10)$$

Where r_Y, r_{Cb} and r_{Cr} are reconstructed residual signals of Y, Cb, Cr components of the block $b_{m,n}$ (8 by 8 pixels). Now for each block, a residual energy that represents actual prediction error is calculated. In the next stage, these blocks are classified according to their residual energies; the sets are reliable, possibly reliable and unreliable. An index number is assigned for each class and motion vector reliability map is constructed, MVRM(m,n). This map is stated in equation 2.11:

$$MVRM(m,n) = \begin{cases} L_1, \text{ If } E_{m,n} \geq \varepsilon_1 \\ L_2, \text{ If neighbor to } L_1 \\ L_3, \text{ otherwise} \end{cases} \qquad (2.11)$$

Where L_1 is the set of unreliable motion vectors L_2 is the set of possible reliable motion vectors L_3 is the set of reliable motion vectors.

2.3.2.2 Macroblock Merging Based on Motion Vector Reliability

At this stage, the connectivity of erroneous motion blocks is investigated. A MB contains 4 motion vectors inside (16 by 16 pixels). The connectivity of MBs is searched in a raster scan order. If the unreliable motion vectors are connected

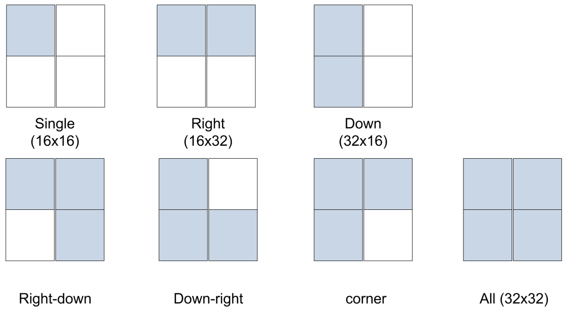
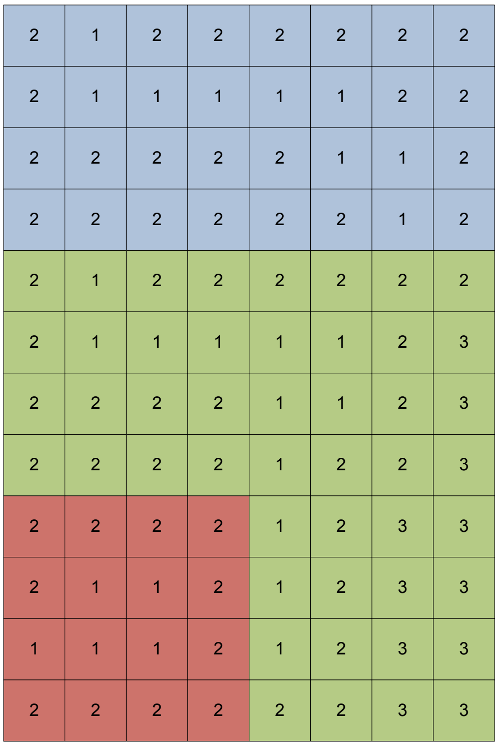


Figure 2.6: MB Merging Shapes



2	1	2	2	2	2	2	2
2	1	1	1	1	1	2	2
2	2	2	2	2	1	1	2
2	2	2	2	2	2	1	2
2	1	2	2	2	2	2	2
2	1	1	1	1	1	2	3
2	2	2	2	1	1	2	3
2	2	2	2	1	2	2	3
2	2	2	2	1	2	3	3
2	1	1	2	1	2	3	3
1	1	1	2	1	2	3	3
2	2	2	2	2	2	3	3

Figure 2.7: MB Merging Examples

as neighbors on the MB boundaries, they are merged. Possible 7 merging shapes are shown in figure 2.6. The upper two MBs are going to be merged because unreliable MVs are connected to each other at the MB boundaries.

1 ⟶ represent an 8 by 8 motion block which has an unreliable MV.

2 ⟶ represent an 8 by 8 motion block which has a possibly reliable MV.

3 ⟶ represent an 8 by 8 motion block which has a reliable MV.

In figure 2.7, some possible MB merging operations are given. The connection of unreliable vectors labeled with 1 is investigated and according to this relation, the blocks are attached to each other. During the following post-processing stages, a merged shape is assumed as one motion block. The merged shapes are shown in same colors.

The diagonal connections of motion blocks which has unreliable motion vectors are skipped because the possibility of belonging to the same object for a diagonal connection has lower probability. A unique number is assigned for a merged MB and a MB merging map is constructed for the next stage, Motion Vector Selection.

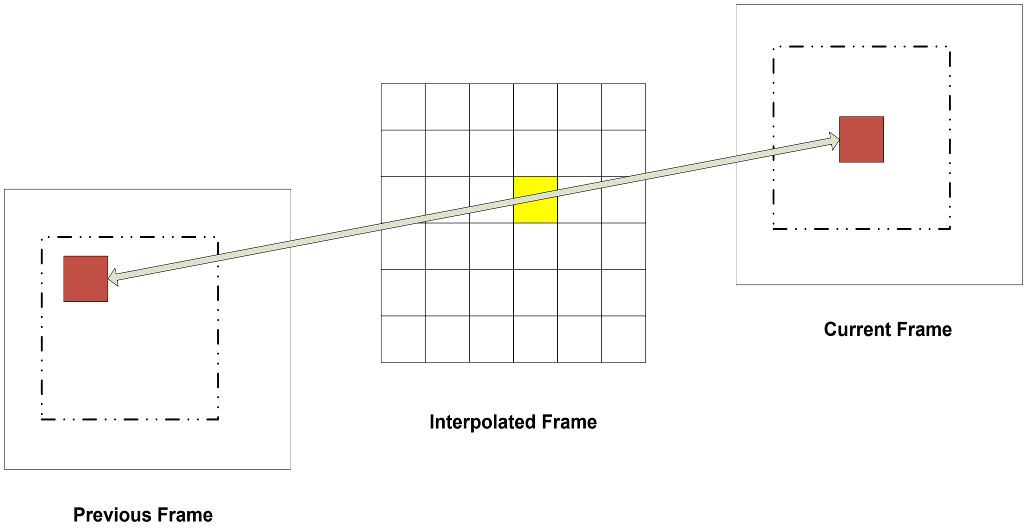


Figure 2.8: Bi-directional Search demonstration

2.3.2.3 Motion Vector Selection

MB merging gives the advantage of correcting unreliable motion vectors in the motion vector selection stage. At this stage, a motion vector which minimizes absolute bidirectional difference for these merged MBs in equations 2.13, 2.12 is searched. Finally, a single vector is assigned to these merged MBs in order to find a globally correct MV for the merged shape. This stage is very important because the motion vectors are estimated according to the frame to be interpolated. Additionally, motion vector selection stage initializes bi-directional search for frame rate interpolation. In figure 2.8, the demonstration of bi-directional search strategy is shown.

$$v_b^* = argmin_{v \epsilon S}(ABPD(v)) \tag{2.12}$$

$$ABPD(v) = \frac{1}{N_G} \sum_{(i,j) \epsilon G} |f_t(i + \frac{1}{2}v_x, j + \frac{1}{2}v_y) - f_{t+1}(i - \frac{1}{2}v_x, j - \frac{1}{2}v_y)| \tag{2.13}$$

Where S denotes the set of motion vector candidates and G denotes the merged group in one of the 7 possible shapes illustrated in figure 2.6.

2.3.2.4 Motion Vector Re-classification Based on Bi-directional Prediction Difference

In the previous stage, for each merged group a best matching global motion vector is assigned according to the ABPD calculation in equation 2.13. However, there may still be smaller areas inside the macroblocks where this new motion vector can not represent their motion well. To find these situations the reliability sets should be reclassified according to their bidirectional difference, in equation 2.14, which is the new energy distribution.

$$BPD_{m,n} = BPD^{Y}_{m,n} + \alpha(BPD^{Cb}_{m,n} + BPD^{Cr}_{m,n}) \tag{2.14}$$

This time two reliability classes are considered; reliable and unreliable. The classification is done according to a new threshold and update the motion vector reliability map (MVRM) in equation 2.15.

$$MVRM(m,n) = \begin{cases} L_1, \text{ If } BPD_{m,n} \geq \varepsilon_3 \\ L_3, \text{ otherwise} \end{cases} \tag{2.15}$$

where L_1 is the set of unreliable MVs, L_3 is the set of reliable MVs. It is important to mention that there are no more possibly reliable MVs. All possibly reliable MVs are classified into L_1 or L_3 in this stage.

2.3.2.5 Motion Vector Refinement

In the previous stage, motion vector field is reclassified as two reliability sets: reliable and unreliable. For unreliable vectors in set L_1, a reliability and similarity constrained vector median filter is applied in equation 2.16:

$$v^{*}_{m,n} = argmin_{v \epsilon S} \sum_{i=m-1}^{m+1} \sum_{j=n-1}^{n+1} w_{i,j} \| v - v_{i,j} \| \tag{2.16}$$

$$w_{i,j} = \begin{cases} \text{If } MVRM(i,j) = L_1, & 0 \\ \text{If } MVRM(i,j) = L_3 \text{ and } d_{i,j} > \varepsilon_4, & 1 \end{cases} \tag{2.17}$$

Where S contains the neighboring motion vectors centered at $v_{m,n}$ and $d_{i,j}$ denotes the distance between $v_{i,j}$ and $v_{m,n}$ using the angular distance measure in 2.18.

$$d_{i,j} = 1 - \frac{v_{m,n} v_{i,j}}{|v_{m,n}||v_{i,j}|} = 1 - \cos(\theta) \tag{2.18}$$

Where θ is the angle between $v_{i,j}$ and $v_{m,n}$. The distance is used for measuring the similarity of the candidate motion vectors and the original motion vector. Two motion vectors are considered to be similar if the distance is below a threshold. This similarity has to be checked because the aim is to identify 8 by 8 blocks having different motion or belonging to another object. Vector median filter sorts the candidate motion vectors that have passed the similarity check and chooses the most probable one. If a MB contains more than half of motion vectors having high difference error energy, motion refinement is not applied in this case because these areas are possibly occlusion regions. If motion refinement is applied on these areas, the structures that have been established by motion vector selection stage may be broken.

In this algorithm, before updating the motion vector of a current block, a check procedure is applied based on the bidirectional prediction difference error of the candidate motion vector. If the error energy is bigger than the error of the original MV, then MV is not updated and a correction may be achieved in the next iteration.

It is important to say that it is not guaranteed to refine all unreliable motion vectors in the set L_1. Some of them may remain unreliable after the motion vector refinement stage. It turns out that most of the remaining unreliable MVs belong to occlusion regions. In the following chapter, we propose an occlusion detection algorithm, and vector assignment and frame synthesis strategy using this occlusion information.

Chapter 3

Occlusion Aware Motion Compensation Scheme for Video Frame Rate Up-Conversion

In this dissertation, we specifically focus on the design of an occlusion artifact free MC technique and propose a new adaptive interpolation scheme using OBMC to remove occlusion artifacts that linear MC techniques can not handle (An example of an occlusion artifact in a MC-FRUC system is presented in figure 3.1). As a result, the proposed MC scheme can cope with large occluded regions and interpolate spatiotemporally consistent pictures.

In sections 3.1 and 3.2, occlusion problem is introduced and in the next section, we define new approaches to solve the problem and improve the performance of the frame rate up-conversion system. In the next chapter, the performance evaluation of the algorithm and comparisons with other methods in the literature will be presented.

3.1 Handling Occlusion Problem

Occlusion is a very common problem in video processing and has many different solutions for each discipline. In the area of stereoscopic imaging, the problem can be solved because you have two pictures which have different point of view of the same scene. With the help of two different views you can reach the depth information of the scene which means the distance information of objects with

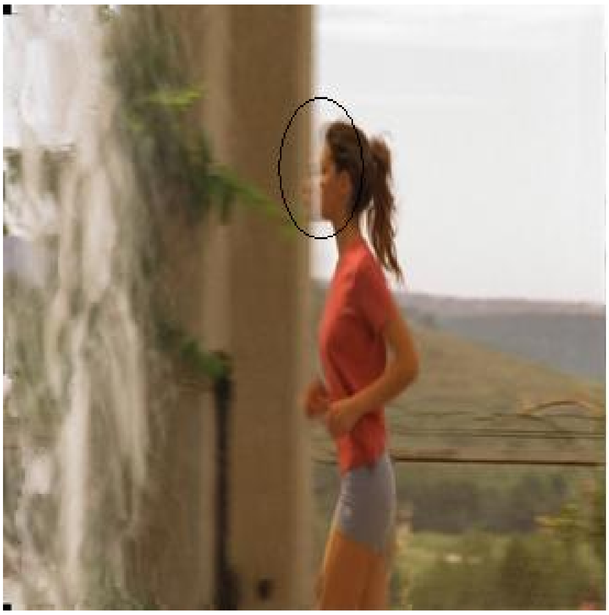


Figure 3.1: Halo artifact caused by occlusion problem

respect to the camera focus. If we combine this information with the estimated motions of objects, it is easy to construct the occlusion information between adjacent frames of a video sequence. Human vision system is an example of a stereoscopic imaging system.

In order to emphasize the importance of the depth information, we can give an example from human eye function which you can experience easily. Think that you and your friend John are passing and catching a basketball between you. Before John throws the ball, close one of your eye with your hand and try to catch the ball with one of your eye. Consequently, you will see that you can not catch the ball with the help of one eye because your brain can not estimate the depth information of the ball and as a result you can not perceive that the ball is coming to you. Likewise the difficulty of this situation, object depth estimation and occlusion detection is a very challenging problem in single-view video. However, there exist methods that can solve the occlusion problem for special cases and help us improve our video processing systems. In the literature, there are three approaches for the occlusion problem. In the following subsections, these methods are described briefly.

3.1.1 Occlusion Detection with Double ME

In this method [2], one backward and one forward motion estimations are applied between two adjacent frames and the errors between them can be stated as follows:

$$SAD_b(\overrightarrow{v}, i, j, t) = \sum_{i,j \epsilon B} |f(i, j, t-1) - f(i + v_x^b, j + v_y^b, t)| \tag{3.1}$$

$$SAD_f(\overrightarrow{v}, i, j, t) = \sum_{i,j \epsilon B} |f(i, j, t) - f(i - v_x^f, j - v_y^f, t-1)| \tag{3.2}$$

The covering/uncovering decision is done according to the relative performance of the forward and backward estimations. The relative performance is calculated as follows:

$$RelPerf = \frac{(SAD_b - SAD_f)}{(SAD_b + SAD_f) + \delta} \tag{3.3}$$

$$OccDecision = \begin{cases} \text{covering,} \quad \text{if } RelPerf \geq Thr \\ \text{uncovering, if } RelPerf \leq -Thr \end{cases} \tag{3.4}$$

where δ is a regularization factor that avoids the decision procedure from making wrong covering/uncovering decisions because of small match errors. It is set to 1% of the maximum error. This method gives us occlusion information between adjacent frames but it is not applicable for picture rate up-conversion due to the fact that the covering/uncovering information for the interpolated frame is uncertain. However, this method can be modified for encoder assisted frame rate conversion, in which frames are intentionally skipped to reduce frame rate, to identify covering/uncovering parts of the skipped frame. In this situation, the encoder may apply forward and backward ME by choosing the skipped frame as reference and construct the exact occlusion information which could be coded and sent as side information to the decoder.

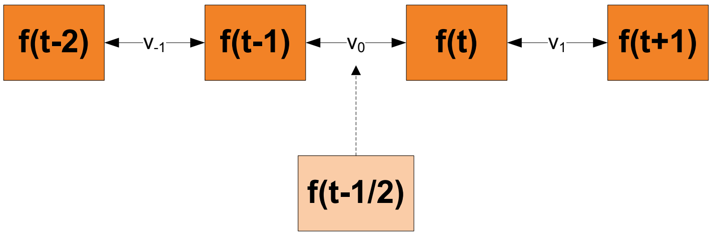


Figure 3.2: Four field occlusion analysis

3.1.2 Occlusion Detection Based on Multiple Frame Analysis

In [8], Thomas and Burl (BBC research group) proposed a method that not only detects the occlusion areas but also assigns correct MVs to these problematic regions. It is based on a four field motion analysis which utilizes two pictures (t-1,t-2) before and two pictures(t,t+1) after the frame being interpolated as shown in figure 3.2. The correct MV assignment is done by calculating three low-pass filtered SADs for each candidate MV as follows:

$$SAD_l(\overrightarrow{V_c}, \underline{x}, t) = \sum_{\underline{x}\epsilon\mathbf{A}} |f(\underline{x} - \frac{3}{2}\overrightarrow{V_c}, t-2) - f(\underline{x} - \frac{1}{2}\overrightarrow{V_c}, t-1)| \qquad (3.5)$$

$$SAD_m(\overrightarrow{V_c}, \underline{x}, t) = \sum_{\underline{x}\epsilon\mathbf{A}} |f(\underline{x} - \frac{1}{2}\overrightarrow{V_c}, t-1) - f(\underline{x} + \frac{1}{2}\overrightarrow{V_c}, t)| \qquad (3.6)$$

$$SAD_r(\overrightarrow{V_c}, \underline{x}, t) = \sum_{\underline{x}\epsilon\mathbf{A}} |f(\underline{x} + \frac{1}{2}\overrightarrow{V_c}, t) - f(\underline{x} + \frac{3}{2}\overrightarrow{V_c}, t+1)| \qquad (3.7)$$

where $\underline{x}$ is the motion block which is the element of the filter window **A**. As you see from the equations the best candidate MV is found and the occlusion regions are weeded out with the following decision:

- **Covered Region :** The SAD of the pictures, $f(t-1)$ and $f(t-2)$, SAD_l, is significantly lower than SAD_m and SAD_r.
- **Uncovered Region :** The SAD of the pictures, $f(t)$ and $f(t+1)$, SAD_r, is significantly lower than SAD_m and SAD_l.

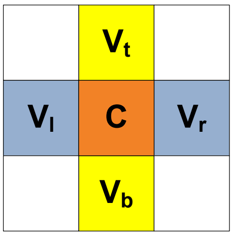


Figure 3.3: Occlusion detection from vector field

In [8], Thomas and Burl uses a complicated weight assignment function that filters the error functions described above and interpolates each pixel using a weighted linear combination of candidates from these four frames. The approaches used in this method seem very attractive but complicated for a real-time application.

3.1.3 Occlusion Detection From Vector Field

In the previous methods, we need to apply extra costly computations before the decision of covering/uncovering. Instead of these operations, the significant discontinuities can be detected as the boundaries of the occlusion area. In [2], a simple method is proposed. In figure 3.3, C represents the current motion block which is desired to be checked whether it is occluded or not and V_l, V_r, V_t and V_b denote the MVs of the neighboring blocks. The discontinuous locations are identified by using simple vector differences as follows:

$$OccDecision = \begin{cases} \text{uncovering}, & \text{if } \overrightarrow{V_l} - \overrightarrow{V_r} \geq Thr \\ \text{covering}, & \text{if } \overrightarrow{V_l} - \overrightarrow{V_r} \leq -Thr \end{cases} \tag{3.8}$$

In our MC-FRUC system, we modified this method and used as an occlusion detector. Therefore, it is very easy to adapt it to our system. The modifications are discussed in section 3.3.

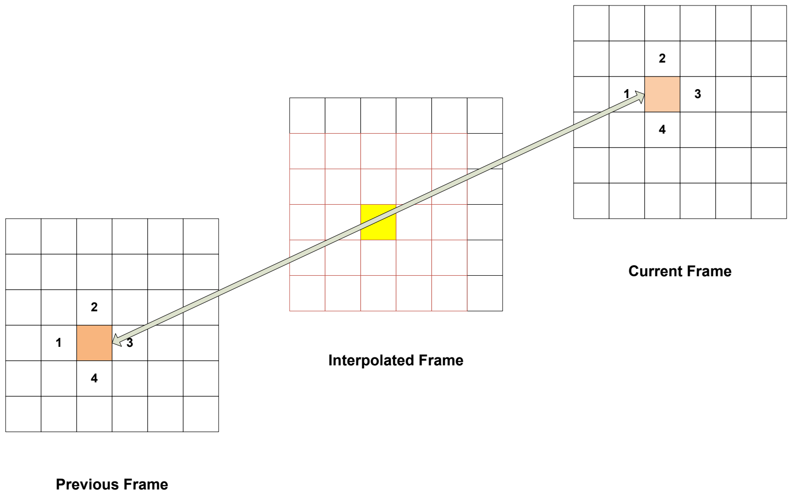


Figure 3.4: Gradient based search for MV smoothing

3.2 Gradient Based Motion Vector Smoothing

After multi-stage motion vector post-processing, we need to refine the motion blocks because some blocks contain object motion boundary inside and these blocks should be divided into sub-blocks to represent the motion of boundaries properly. In [4] Nguyen proposed a motion vector smoothing method that uses a fixed averaging filter kernel to smooth out the motion field. In this thesis, we propose a new motion vector smoothing method which adapts itself to refine the motion of object boundaries. In order to detect the object boundary, we define an error criteria that modifies the previously discussed absolute bi-directional prediction difference (ABPD). ABPD criteria has higher values especially on the object boundaries because there occurs high frequency changes between object boundary and the background. Due to this reason, we are faced with problems to distinguish the background and object boundary pixels. If ABPD is normalized by its local gradient, this approach provides us to detect the high frequency changes along the motion boundary and minimizing this error help us to find the true motion vector for the pixels on the object boundary. In figure 3.4, we demonstrate the gradient based search for motion vector smoothing. We search for the best motion vector that represents the true motion. During the search operation, we decrease the block size from 8 × 8 to 4 × 4. For each 4 × 4 block we calculate the corresponding gradient value as follows:

$$G_p(v_i) = \sqrt{[med(B_1^p) - med(B_3^p)]^2 + [med(B_2^p) - med(B_4^p)]^2} \tag{3.9}$$

$$G_n(v_i) = \sqrt{[med(B_1^n) - med(B_3^n)]^2 + [med(B_2^n) - med(B_4^n)]^2} \tag{3.10}$$

where G_p and G_n represent block gradient values on previous and next frames respectively, B_i^p and B_i^n represent the neighboring 4 × 4 blocks in previous and next frames and *med()* operator takes the median value of 4 pixels in a 4 × 4 motion block. Now we can define our new error criteria gradient normalized ABPD value as follows:

$$GradABPD(v_i) = \frac{ABPD(v_i) + const}{\frac{1}{2}G_p(v_i) + \frac{1}{2}G_n(v_i) + const} \tag{3.11}$$

Where

$$ABPD(v_i) = \frac{1}{N_B} \sum_{(i,j)\epsilon B} |f_t(i + \frac{1}{2}v_i^x, j + \frac{1}{2}v_i^y) - f_{t+1}(i - \frac{1}{2}v_i^x, j - \frac{1}{2}v_i^y)| \tag{3.12}$$

In equation 3.12, B represents a 4 × 4 block.

The behavior of this error function is adaptive to the edges. On smooth regions it behaves like normal ABPD function. On the other hand, at object boundaries the denominator becomes more dominant and avoids a matching between smooth background and detailed object. In other words, for detailed blocks this new error function favors candidate blocks that also have high gradient.

The best motion vector is chosen according to the minimum value of this error criteria. The search range of the smoothing stage is shown as red grid (5 × 5 search window) in figure 3.4.

$$v_b^* = argmin_{v_i\epsilon S}(GradABPD(v_i)) \tag{3.13}$$

3.3 Occlusion Detection for Motion Compensated Frame Rate Up-Conversion

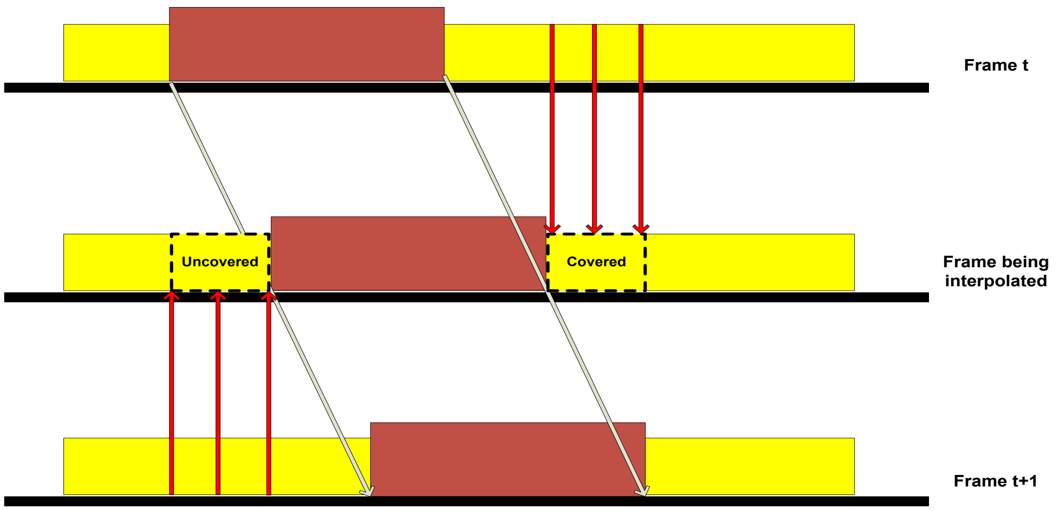


Figure 3.5: Covered and uncovered handling in MC-FRUC

After the MV post-processing and gradient based smoothing, we get a more reliable MV field that represents true motion information between adjacent frames. However, there are still MBs which have high error energy; that is, wrong MVs are estimated for the covered and uncovered regions. For example, if we consider an uncovering case; there is a background MB which doesn't exist in previous frame and it suddenly appears in the next frame; both covering and uncovering scenario are demonstrated for MC-FRUC in figure 3.5. Accordingly, ME and MV post-processing techniques can not find a best match for this part of the image. As a consequence there should be a motion representation for these problematic regions in order to interpolate visually good quality images between adjacent frames. In brief, we should detect and identify covered and uncovered regions and then at the interpolation stage reliable MVs should be estimated for these regions. In this section, an occlusion detection stage is presented. In general, if a MB has a high $GradABPD$ value, this is a strong sign that this MB is inside an occluded region. To determine whether the region is covered or uncovered, we suggest to look at the neighboring MV differences. For instance, suppose that in the above figure MB (k=0) is an occluded MB. If the left and right neighboring

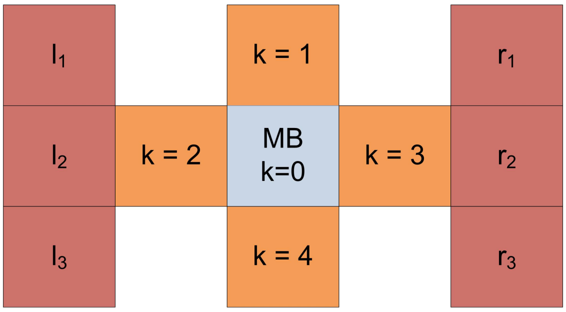


Figure 3.6: The demonstration of adjacent MBs for Occlusion Detection

blocks have a positive MV difference then this MB is going to be covered; otherwise MB is going to be uncovered. To make sure that these neighboring blocks are representative of the different motions in the region, they are chosen slightly further away to the left and right of the current MB. The decision algorithm can be summarized as follows:

Covered - Uncovered Decision Algorithm

```
if ($GradABPD_{m,n} > \varepsilon_5$) //this MB is occluded
if $med(\{v^x_{r_i}\}_{1 \leq i \leq 3}) > med(\{v^x_{l_i}\}_{1 \leq i \leq 3})$, then $K_{m,n} = 1$ //Covered
if $med(\{v^x_{r_i}\}_{1 \leq i \leq 3}) < med(\{v^x_{l_i}\}_{1 \leq i \leq 3})$, then $K_{m,n} = 0$ //Uncovered
else $K_{m,n} = 0.5$ // neither covered nor uncovered
```

In the above algorithm, ε_5 is a predefined and experimentally found occlusion threshold, m, n represents the current MB that is probably occluded, $\{v^x_{r_i}\}_{1 \leq i \leq 3}$ is the array containing horizontal MV components of one block far right neighbors' MVs shown in 3.6 as r_i and $\{v^x_{l_i}\}_{1 \leq i \leq 3}$ is the array containing the horizontal MV components of the one block far left neighbors' MVs shown in figure as l_i. A similar decision can be carried out for vertical motions using the up and down neighbors and y components of their MVs. If the MB is covered, we identify it on the occlusion map, $K_{m,n}$, as 1. If it is uncovered, we identify it on the occlusion map as 0 and if neither covered nor uncovered is the case, we identify this MB as 0.5. In order to generate a smooth occlusion map, we filter the map with a 7x7 H^{avg}_{7x7} smoothing moving average filter as follows:

$$SmoothK = K * H^{avg}_{7x7} \tag{3.14}$$

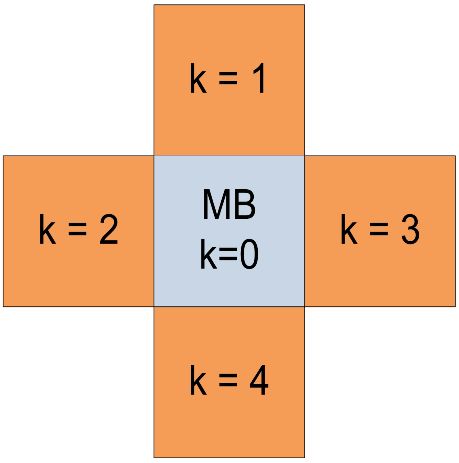


Figure 3.7: The demonstration of adjacent MBs for Overlapped-Block Motion Compensation

3.4 An Occlusion Adaptive Overlapped Block Motion Compensation

The standard linear MC frame synthesis method creates annoying artifacts because of the inconsistencies in the motion field, and the difficulty of estimating correct MVs in occluded regions. To improve the consistency of the synthesized frame, overlapped block motion compensation (OBMC) can be applied. OBMC is a useful MC technique which reduces the blocking artifacts caused by conventional block based video coders [13]. Therefore, OBMC provides visually good quality pictures for MC-FRUC problem. In OBMC, each MB is synthesized as weighted average of multiple predictions using MVs of both the current MB and its neighboring MBs like in figure 3.7. The weights are chosen inversely proportional to gradient based absolute bi-directional prediction difference (Grad-ABPD) of corresponding MVs. If the video sequences consist of large motion, the occluded regions get bigger and during the MC we are faced with annoying hollow artifacts around moving object boundaries. In the previous stage, we have detected the covered/uncovered regions. With the help of the occlusion detector, the right frame is chosen for the occluded MB. The adaptation is done using the occlusion map as a weighting factor between adjacent frames. In figure 3.7, k represents adjacent MB index and v_k is the estimated MV for the k^{th} adjacent MB. $GradABPD$ is calculated for MB (m, n) using the neighbouring MV v_k. A weight is assigned for the MV v_k as follows:

$$w_k = GradABPD(v_k)_{m,n}^{-1} \tag{3.15}$$

S_w is the normalization factor so that the summation of the weights should be 1.

$$S_w = \sum_{k=0}^{4} w_k \tag{3.16}$$

For each MV v_k we generate a candidate block of pixels for MB (m, n). The occlusion map smoothly switches between adjacent frames according to the occlusion type.

$$C_k = SmoothK_{m,n} f_{t-1}(i+\frac{v_k^x}{2}, j+\frac{v_k^y}{2})+(1-SmoothK_{m,n}) f_t(i-\frac{v_k^x}{2}, j-\frac{v_k^y}{2}) \tag{3.17}$$

Finally, the interpolated mid frame is generated as the weighted sum of candidate MBs C_k

$$f_{t-\frac{1}{2}}(i, j) = \sum_{k=0}^{4} \frac{w_k}{S_w} C_k \tag{3.18}$$

The block diagram of the overall MC-FRUC system is illustrated in figure 3.8

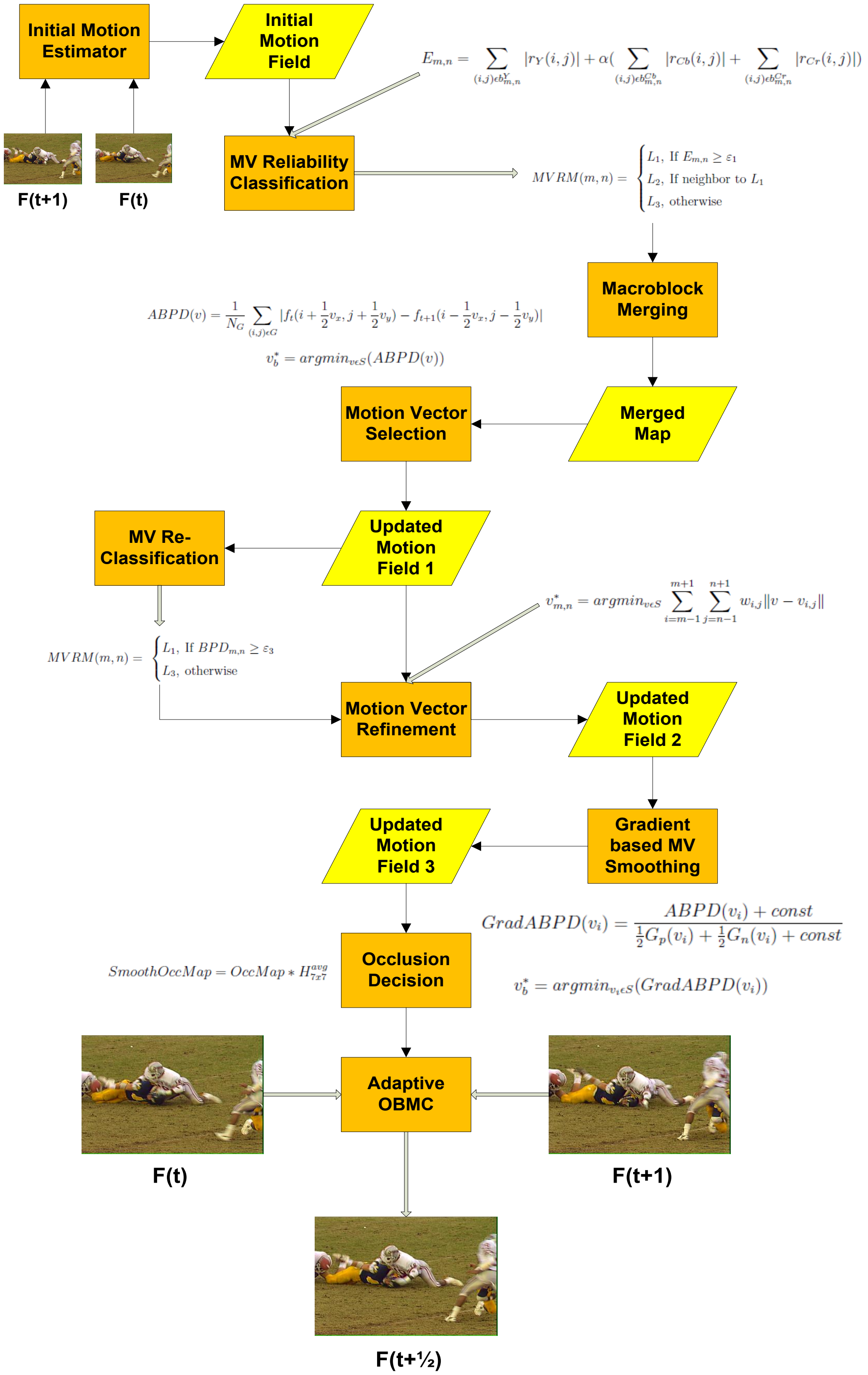

Initial Motion Estimator
Initial Motion Field
F(t+1)
F(t)
$E_{m,n} = \sum_{(i,j)\epsilon b^{Y}_{m,n}} |r_Y(i,j)| + \alpha(\sum_{(i,j)\epsilon b^{Cb}_{m,n}} |r_{Cb}(i,j)| + \sum_{(i,j)\epsilon b^{Cr}_{m,n}} |r_{Cr}(i,j)|)$
MV Reliability Classification
$MVRM(m,n) = \begin{cases} L_1, \text{ If } E_{m,n} \geq \varepsilon_1 \\ L_2, \text{ If neighbor to } L_1 \\ L_3, \text{ otherwise} \end{cases}$
Macroblock Merging
$ABPD(v) = \frac{1}{N_G} \sum_{(i,j)\epsilon G} |f_t(i + \frac{1}{2}v_x, j + \frac{1}{2}v_y) - f_{t+1}(i - \frac{1}{2}v_x, j - \frac{1}{2}v_y)|$
$v_b^* = argmin_{v\epsilon S}(ABPD(v))$
Motion Vector Selection
Merged Map
MV Re-Classification
Updated Motion Field 1
$v^*_{m,n} = argmin_{v\epsilon S} \sum_{i=m-1}^{m+1} \sum_{j=n-1}^{n+1} w_{i,j} \|v - v_{i,j}\|$
$MVRM(m,n) = \begin{cases} L_1, \text{ If } BPD_{m,n} \geq \varepsilon_3 \\ L_3, \text{ otherwise} \end{cases}$
Motion Vector Refinement
Updated Motion Field 2
Updated Motion Field 3
Gradient based MV Smoothing
$GradABPD(v_i) = \frac{ABPD(v_i) + const}{\frac{1}{2}G_p(v_i) + \frac{1}{2}G_n(v_i) + const}$
Occlusion Decision
$SmoothOccMap = OccMap * H^{avg}_{7x7}$
$v_b^* = argmin_{v_i\epsilon S}(GradABPD(v_i))$
Adaptive OBMC
F(t)
F(t+1)
F(t+½)

Figure 3.8: The Overall Frame Rate Up-Conversion System

Chapter 4

Simulation Results and Comparisons

In this chapter, simulation results and comparisons with other algorithms are presented. In our simulations we prefer to test our algorithm on commonly used test video sequences in the literature. The video quality measurement has been an important research subject and the properties of the human vision system is the first issue to understand in this research. There isn't a perfect analytic subjective error measure for video processing due to the fact that everybody has his/her own tastes.

Researchers have been using some subjective and objective testing techniques. Objective measures are commonly used for video coding but not so useful for FRUC problem. Objective measures may not be adequate to assess the perceptual quality of the video sequence. Even though the objective and the subjective quality measures are consistent in typical cases, they may also contradict each other for some cases. For meaningful quality assessment, it is important to understand the different factors that affect the visual perception of the video.

The video enhancement research focuses on the tastes of the human vision system. If you present a better quality taste to people, they can distinguish the difference between past and present. HDTV is a nice example for this issue, after the awareness of high definition quality people expect the same quality from the older recorded videos. According to these needs video enhancement is going to be an open research subject in video processing.

4.1 Objective Quality Measures for Video Processing

4.1.1 Peak Signal To Noise Ratio (PSNR)

It is the ratio between the maximum power of a signal and the power noise corrupting the image. PSNR is expressed in terms of logarithmic decibel scale. The PSNR metric is the most commonly preferred quality measure for lossy compression. The signal is the original raw data and the noise is the error that is caused by a manipulation of the original signal. A lower PSNR value indicates that the quality of the measured data is lower, a higher PSNR value indicates higher quality. It is defined via mean squared error (MSE):

$$MSE = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [f(i,j) - f_{noisy}(i,j)]^2 \tag{4.1}$$

Then we can define PSNR as:

$$PSNR = 10log_{10}(\frac{MAX_I^2}{MSE}) \tag{4.2}$$

where MAX_I is the maximum possible pixel value of the image. If the pixels are represented by 8 bits per sample, this is 255.

4.1.2 Structural Similarity Test (SSIM)

The structural similarity [14] is a quality measurement method for measuring the similarity between two images. The similarity task is separated into two comparisons: Luminance and contrast. To compare the luminance of each signal component, we used the estimate of the local mean intensity:

$$\mu_x = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{4.3}$$

Where x is the array that contains the pixel values of a block. The standard deviation can be used as an estimate of the signal contrast:

$$\sigma_x = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)^2} \tag{4.4}$$

The tests are block based tests and compare the images block by block. In equations 4.5, 4.6 and 4.7 x represents the block on the original image and y is the corresponding block on the processed image. For luminance comparison we define:

$$I(x,y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} \tag{4.5}$$

For contrast comparison we define:

$$C(x,y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2} \tag{4.6}$$

It is important to mention that these two comparisons are independent from each other. Finally, we can define the overall SSIM metric as follows:

$$SSIM(x,y) = I(x,y)C(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_x\sigma_y + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{4.7}$$

where

- $c_1 = (k_1L)^2$, $c_2 = (k_2L)^2$ two variables to stabilize the division with weak denominator.
- L is the dynamic range of the pixel values $(2^{numberofbitsperpixel} - 1)$
- $k_1 = 0.01$ and $k_2 = 0.03$ by default

In order to evaluate the image quality this formula is applied only on the luminance. The resultant SSIM index is a decimal value between -1 and 1, and value 1 is only reachable in the case of two identical sets of data. Typically it is calculated on window sizes of 8 by 8 (motion blocks). The window can be displaced pixel-by-pixel on the image but in the literature researchers prefer to use only a subgroup of the possible windows to reduce the complexity of the calculation and take the average value of SSIM on the overall image.

4.2 Subjective Quality Measure for Video Processing

4.2.1 Subjective Mean Squared Error (SMSE)

In the purpose of studying the quantitative relation between MSE and perceived similarity, subjective mean squared error criteria is proposed by Hans Marmolin [15]. The aim of this research is to derive more valid error measures by weighting the error function in accordance with assumed properties of the human vision system. Let D_i represents the error function and it is defined as follows:

$$D_i = \frac{(M(x_i) - M(y_i)) \times (1 + G(x_i))}{1 + 2 \times S(x_i)} \tag{4.8}$$

- Where x represents the original frame and i is the pixel index. Likewise, y represents the frame which is being tested.
- M is the mean value of pixels in a 3×3 window surrounding the original and processed picture.
- $G(x_i)$ is the gradient level of pixel i in the original picture.
- $S(x_i)$ is the standard deviation in a 7×7 window surrounding pixel i in the original image.

SMSE is a nice error measure for video frame rate up-conversion system because it weights the errors on the object motion boundaries more heavily than distortions in other areas of the picture. This property is a strong sign that SMSE is a good approximation of human vision system. For example, in textured areas this error function is lower due to high standard deviation, which is compatible by the fact that human eye can not detect the errors in textured regions. On the other hand, in a smooth surface human eye can detect errors more easily; hence the error function gives a higher value because the smooth surface has a lower standard deviation. Average SMSE value is calculated as follows:

$$SMSE = [\frac{1}{n}\sum_{i=1}^{n} |D_i^2|^p]^{\frac{1}{p}} \tag{4.9}$$

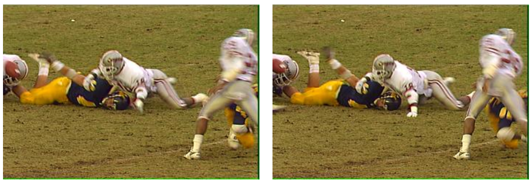
Where p is a factor that determines the relative importance of large and small errors.

4.3 Simulation Results

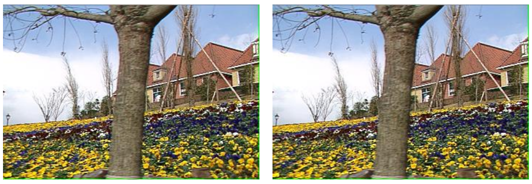
Figure 4.1 shows the previous and next frames from football and flower garden sequences that are used to interpolate the missing intermediate frames. In figures 4.2 and 4.4, we present the results of adaptive OBMC in [6] which used a different adaptive synthesis strategy. In figures 4.3 and 4.5, it is obvious that our proposed OBMC scheme generates fewer artifacts than this algorithm. Hence, occlusion aware processing improves the interpolation quality in video frame rate up-conversion problem. The halo artifact is reduced in our interpolated pictures, however in AOBMC scheme [6] the repeated parts of the object boundaries can be seen which is an occlusion artifact.

In figure 4.6, we present the result of another adaptive interpolation scheme [7] that uses occlusion decision. When we compare the proposed OBMC scheme and adaptive FRC [7], it can be concluded that our algorithm achieves better motion classification and also the success of OBMC over linear motion compensation is evident from the results. Our algorithm performs better occlusion decision and the difference can be seen on the body of the tree shown as a red circle in figure 4.6 when compared to figure 4.7.

Football sequence frame 71 Football sequence frame 72



Football sequence frame 3 Football sequence frame 4



Flower garden sequence frame 11 Flower garden sequence frame 12

Figure 4.1: Tested video frames

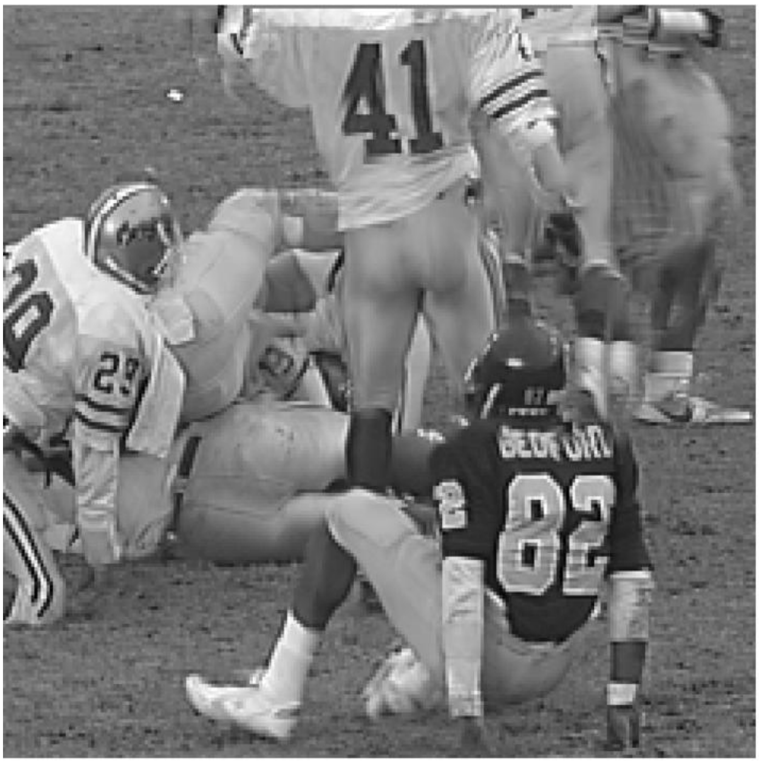


Figure 4.2: Adaptive OBMC [6]

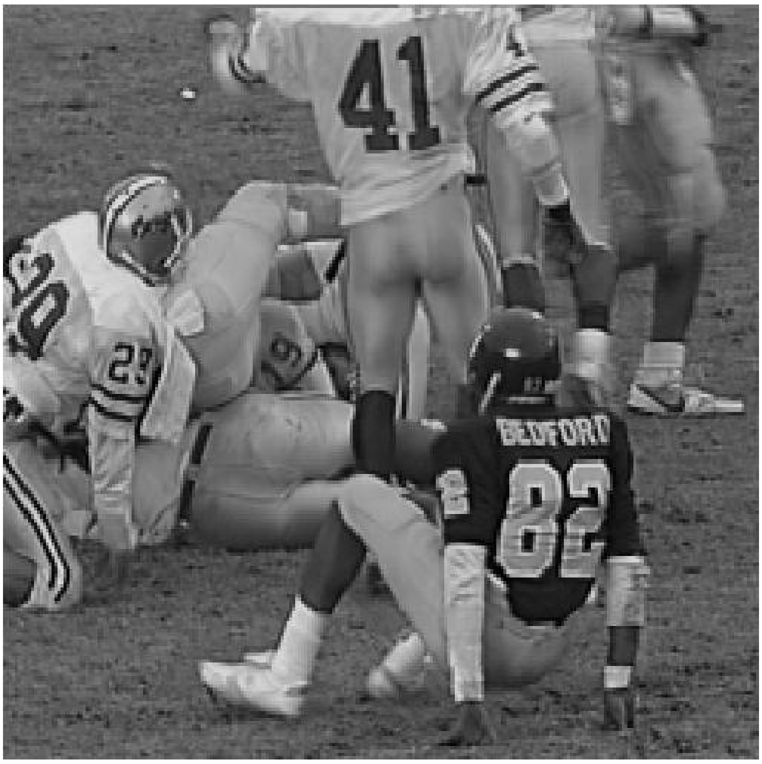


Figure 4.3: The proposed Occlusion aware OBMC scheme

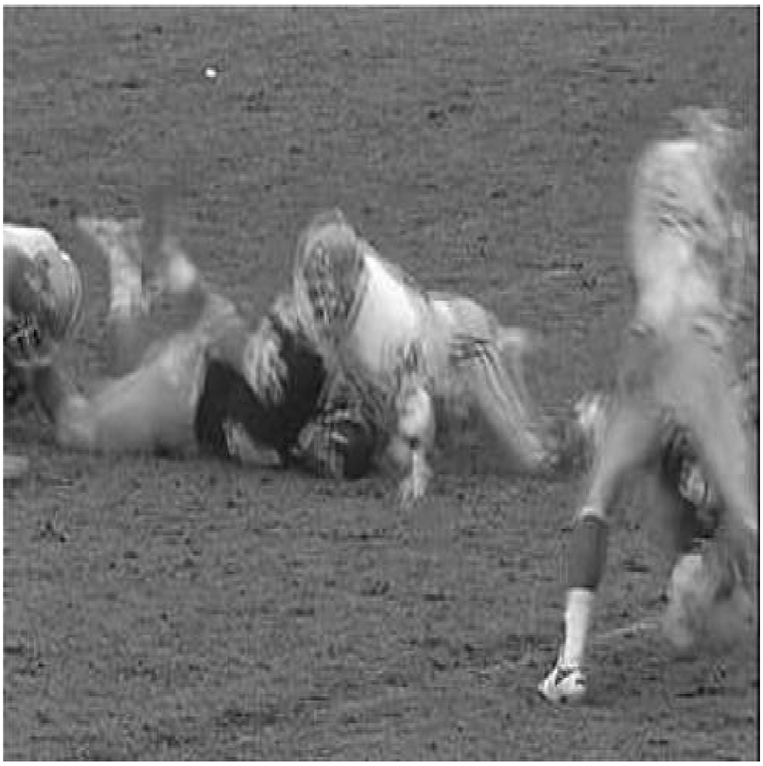


Figure 4.4: Adaptive OBMC [6]

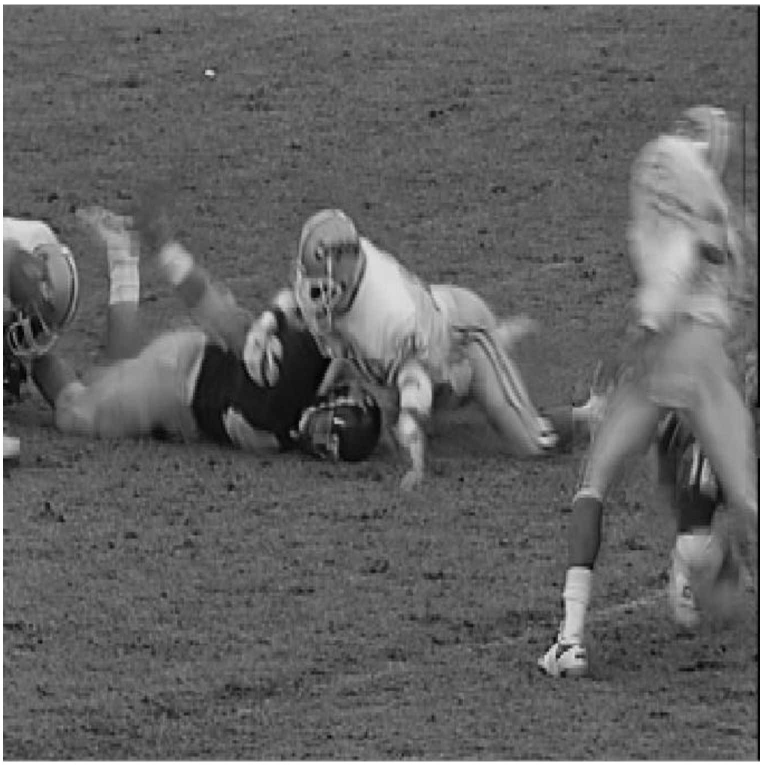


Figure 4.5: The proposed Occlusion aware OBMC scheme

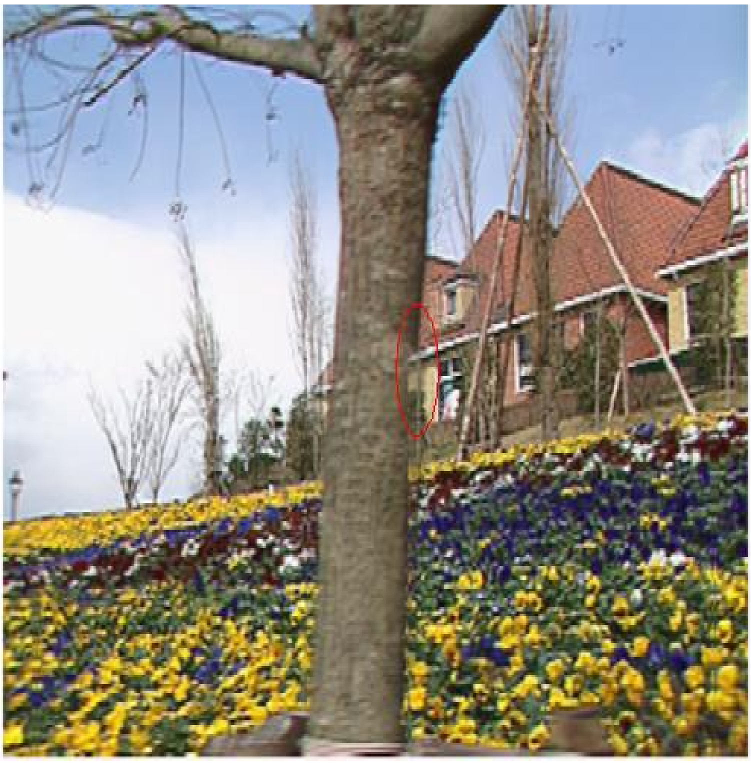


Figure 4.6: Adaptive Frame Rate Conversion [7]

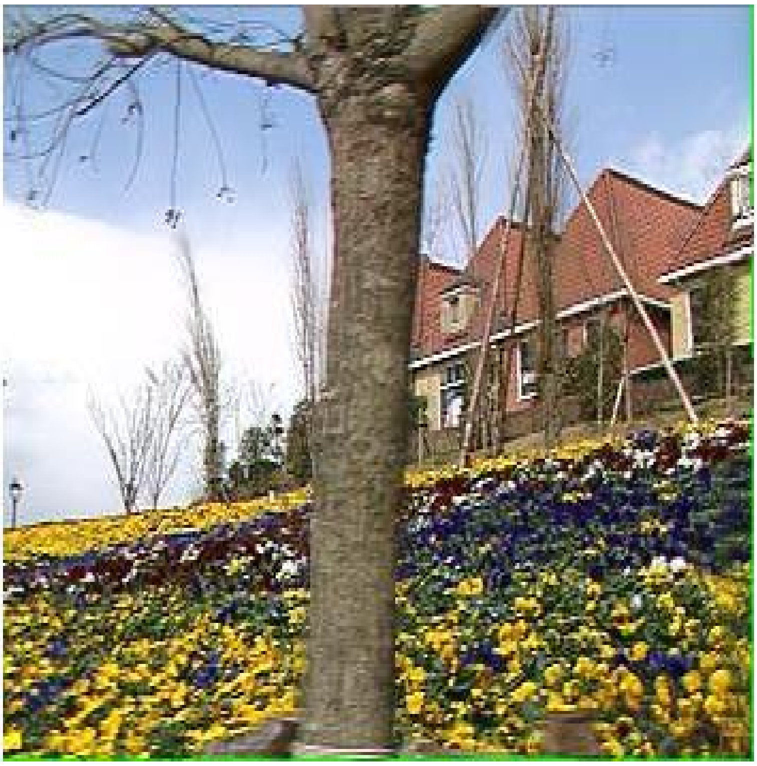


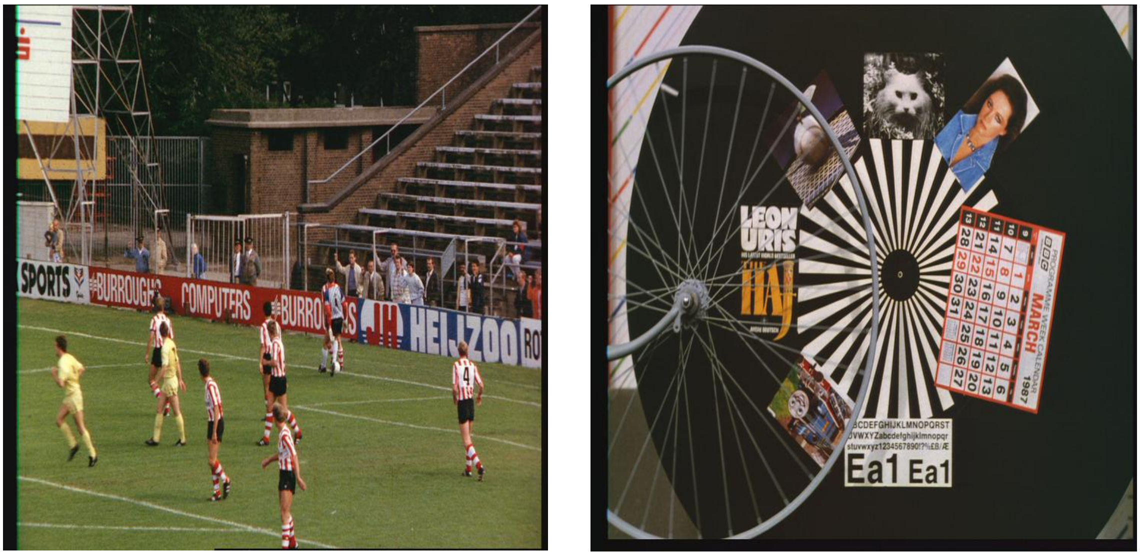
Figure 4.7: The proposed Occlusion aware OBMC scheme

The objective measurements on tables 4.1, 4.2, 4.3, 4.4, 4.5 , 4.6 are done on the Y component (only luminance) of the video signals. During the simulations, we skipped the even numbered frames and interpolate the missing frame using the method mentioned above the tables.

In table 4.1, we compare our algorithm with another post-processing method, vector median filtering [10] using PSNR metric. The test videos include rotation and zooming motions which are difficult to estimate but the proposed MC has better performance as seen on table 4.1.

Table 4.1: PSNR Comparison with [10]

Video	**vec.lin**	**vec.med3**	**vec.CWM**	**OBMC+OCC**
Football	26.34	25.66	28.49	28.82
Wheel	24.55	23.37	27.6	28.47



Football sequence (zooming)

Wheel sequence (rotation)

Figure 4.8: Test sequences on table 4.1

The abbreviations for the operations of the algorithms are as follows:

- LinearMC : Bi-directional linear motion compensation is used.
- OBMC : Standard OBMC is used.
- OBMCsm : OBMC is used with gradient based smoothing.
- OCC : Occlusion decision is applied.
- HS : Hexagonal search is used as initial motion estimator.
- 3DRS : 3DRS algorithm is used as initial motion estimator.

In table 4.2, the objective performance of Hexagonal search and 3-D recursive search is tested in the algorithm. During the simulations, both ME types are post-processed by multi-stage MV post-processing algorithm in [4], gradient based smoothing, occlusion detection stage and OBMC are applied in order. As seen on table 4.2, neither HS nor 3DRS has a dominant performance in our simulations.

Table 4.2: HS and 3DRS performance comparison

Videos	**HS**		**3DRS**	
	PSNR	**SSIM**	**PSNR**	**SSIM**
Akiyo	45,04	**0,996**	**45,37**	**0,996**
Salesman	41,40	**0,991**	**41,67**	**0,991**
Carphone	30,33	**0,921**	**30,69**	**0,921**
Bus	24,47	0,866	**24,73**	**0,874**
Flower Garden	**24,26**	**0,905**	24,08	0,899
Football 1	21,66	**0,755**	**21,29**	0,737
Foreman	**30,83**	**0,912**	30,57	0,908
Mobile Calendar	**26,14**	**0,919**	26,12	**0,919**
Mother	39,22	**0,970**	**39,29**	**0,970**
News	**34,62**	**0,976**	34,60	**0,976**
Stefan	23,93	0,829	**24,22**	**0,851**
Table Tennis	**27,50**	**0,926**	27,31	0,924

Videos	**HS**		**3DRS**	
	MSE	**SMSE**	**MSE**	**SMSE**
Akiyo	3,35	0,85	**3,18**	**0,82**
Salesman	5,75	1,17	**5,52**	**1,12**
Carphone	86,42	4,76	**84,20**	**4,60**
Bus	240,34	8,18	**225,63**	**7,80**
Flower Garden	**259,31**	**6,54**	269,82	6,68
Football 1	**451,14**	**10,62**	491,24	11,02
Foreman	**66,69**	**4,10**	69,99	4,22
Mobile Calendar	**186,71**	**5,81**	187,46	5,82
Mother	8,69	1,66	**8,68**	**1,65**
News	**31,06**	2,81	31,15	**2,79**
Stefan	345,41	8,64	**294,78**	**7,97**
Table Tennis	**194,18**	**5,07**	196,69	5,14

In tables 4.3 and 4.4 intermediate frames are interpolated by using bi-directional linear MC, standard OBMC, occlusion adaptive OBMC and the proposed gradient based MV smoothing with occlusion adaptive OBMC scheme. The objective metrics are computed using the skipped frames as original ones and applying the metric computations as discussed in the previous section. The best value for a test video is typed in bold font. From tables 4.3 and 4.4 it is misunderstood that bi-directional linear MC and standard OBMC without occlusion decision stage achieve better quality. However, subjective comparisons contradict these values. This contradiction is demonstrated in figure 4.9.

Table 4.3: PSNR and SSIM measurements

Videos	**OBMC**		**OBMC+OCC**		**OBMCsm+OCC**	
	PSNR	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**
Akiyo	46,30	0,996	46,30	0,996	45,04	0,996
Salesman	42,10	0,991	42,05	0,991	41,40	0,991
Carphone	30,33	0,920	30,28	0,920	30,33	0,921
Bus	**24,81**	**0,867**	24,71	0,866	24,47	0,866
Flower Garden	**25,87**	**0,920**	25,73	0,919	24,26	0,905
Football 1	**21,84**	**0,760**	**21,84**	**0,760**	21,66	0,755
Foreman	30,23	0,908	30,18	0,908	**30,83**	**0,912**
Mobile Calendar	**26,26**	**0,920**	26,11	0,918	26,14	0,919
Mother	39,82	**0,970**	39,80	**0,970**	39,22	**0,970**
News	34,82	0,975	34,57	0,975	34,62	**0,976**
Stefan	23,80	0,831	**23,82**	**0,832**	23,93	0,829
Table Tennis	**28,97**	**0,935**	28,75	0,934	27,50	0,926

Videos	**LinearMC**		**LinearMC+OCC**	
	PSNR	**SSIM**	**PSNR**	**SSIM**
Akiyo	**49,17**	**0,998**	**49,17**	**0,998**
Salesman	**43,42**	**0,992**	**43,42**	**0,992**
Carphone	**31,78**	**0,928**	**31,78**	**0,928**
Bus	19,61	0,491	19,61	0,491
Flower Garden	17,71	0,535	17,70	0,535
Football 1	20,19	0,676	20,19	0,676
Foreman	29,95	0,852	29,95	0,852
Mobile Calendar	26,42	0,907	26,41	0,906
Mother	**40,65**	0,965	**40,65**	0,965
News	**35,13**	0,974	35,11	0,974
Stefan	21,23	0,681	21,13	0,679
Table Tennis	24,18	0,783	24,15	0,783

Table 4.4: MSE and SMSE measurements

Videos	**OBMC**		**OBMC+OCC**		**OBMCsm+OCC**	
	MSE	**SMSE**	**MSE**	**SMSE**	**MSE**	**SMSE**
Akiyo	3,03	0,72	3,03	0,72	3,35	0,85
Salesman	5,09	0,98	5,20	1,00	5,75	1,17
Carphone	88,06	4,72	89,47	4,77	86,42	4,76
Bus	**221,95**	**7,74**	226,84	7,84	240,34	8,18
Flower Garden	**172,11**	**5,40**	180,19	5,48	259,31	6,54
Football 1	**433,10**	**10,61**	**433,10**	**10,61**	451,14	10,62
Foreman	80,77	4,33	81,91	4,36	**66,69**	4,10
Mobile Calendar	186,06	5,80	190,97	5,93	186,71	5,81
Mother	**7,84**	1,45	7,92	1,46	8,69	1,66
News	**30,71**	2,73	32,00	2,83	31,06	2,81
Stefan	359,86	8,72	350,79	8,70	**345,41**	**8,64**
Table Tennis	**130,85**	**4,39**	137,39	4,60	194,18	5,07

Videos	**LinearMC**		**LinearMC+OCC**	
	MSE	**SMSE**	**MSE**	**SMSE**
Akiyo	**1,30**	**0,36**	**1,30**	**0,36**
Salesman	**4,04**	**0,74**	**4,04**	**0,74**
Carphone	**70,51**	**3,90**	**70,51**	**3,90**
Bus	738,75	13,84	738,75	13,84
Flower Garden	1127,74	12,10	1129,45	12,13
Football 1	630,80	11,71	630,80	11,71
Foreman	111,15	4,02	111,15	**4,02**
Mobile Calendar	**184,69**	**4,32**	185,39	4,31
Mother	8,07	**1,17**	8,09	1,18
News	32,76	**2,40**	34,13	2,43
Stefan	620,31	11,31	635,72	11,51
Table Tennis	497,68	6,59	502,16	6,68

In table 4.5, the method in reference [9], Algorithm 1, is compared objectively with our proposed method. Although it seems that the performance of our algorithm looks lower than Algorithm 1, it is important to mention that in [9] the MC-FRUC algorithm skips the interpolation procedure if the estimation error is larger than a threshold. In our simulations, we never skip the interpolation procedure and take into account large errors in our measurements.

Table 4.5: Performance comparison with Algorithm 1

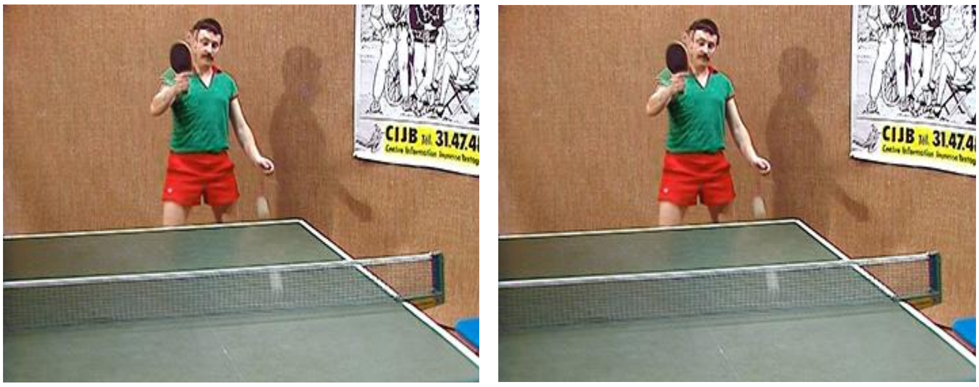
Videos	**Algorithm 1**		**OBMCsm (3DRS)**	
	PSNR	**SSIM**	**PSNR**	**SSIM**
Akiyo	39,40	0,968	**45,37**	**0,996**
Salesman	34,30	0,913	**41,67**	**0,991**
Carphone	**33,36**	0,913	30,69	**0,921**
Flower Garden	**29,36**	**0,931**	24,08	0,899
Football 1	**28,05**	**0,814**	21,29	0,737
Mobile Calendar	**30,73**	**0,949**	26,12	0,919
Mother	37,37	0,935	**39,29**	**0,970**
Stefan	**29,60**	**0,896**	24,22	0,851
Table Tennis	**32,53**	0,869	27,31	**0,924**

In table 4.6, we compare our experiments with the latest paper of Nguyen, [5]. Just like Algorithm 1, Nguyen also skips the interpolation procedure if the occlusion size gets bigger which implies a high average error during the interpolation. It is very obvious that our performance measures are quite close despite the fact that we never skip interpolation and take into account all interpolated frames to show the system performance objectively.

Table 4.6: Performance comparison with journal [5]

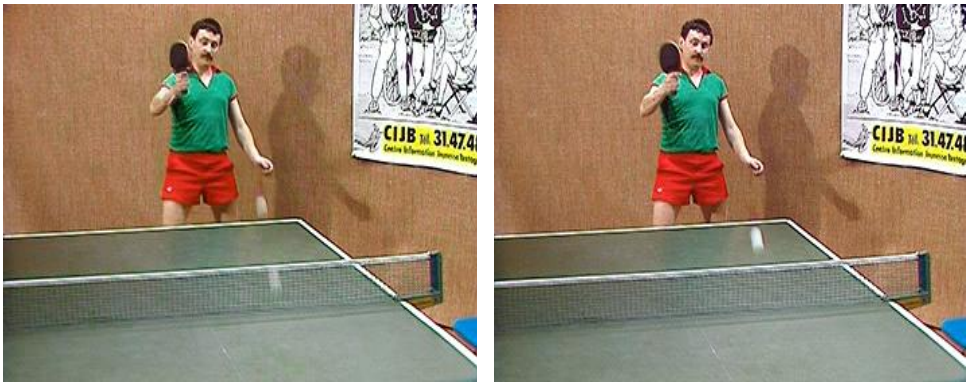
Videos	**Bi-Pred**		**Adaptive-Pred**		**OBMCsm (3DRS)**	
	PSNR	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**
Football 1	**25,11**	**0,780**	24,25	0,760	21,29	0,737
Foreman	**31,70**	**0,960**	31,52	**0,960**	30,57	0,908
Stefan	**26,49**	**0,900**	25,93	0,890	24,22	0,851

In figure 4.9, the 46th frame of the Table Tennis video sequence is interpolated using Linear MC, OBMC and occlusion adaptive OBMC with gradient based smoothing. As seen on figure 4.9, the right ear of the man has serious artifacts in cases Linear MC and OBMC. In contrast, occlusion adaptive OBMC with gradient based smoothing has better visual quality subjectively. However, the objective measure PSNR gives us wrong information in this case. Similar to [5] the objective experiments are in contradiction with the subjective picture quality. In conclusion, it is important to mention that PSNR, SSIM, MSE and SMSE objective error measures are not reliable to measure the quality of the interpolated pictures in video frame rate up-conversion problem.



Linear MC
(PSNR:31.902 dB, SSIM:0.967)

OBMC
(PSNR:31.900 dB, SSIM:0.966)

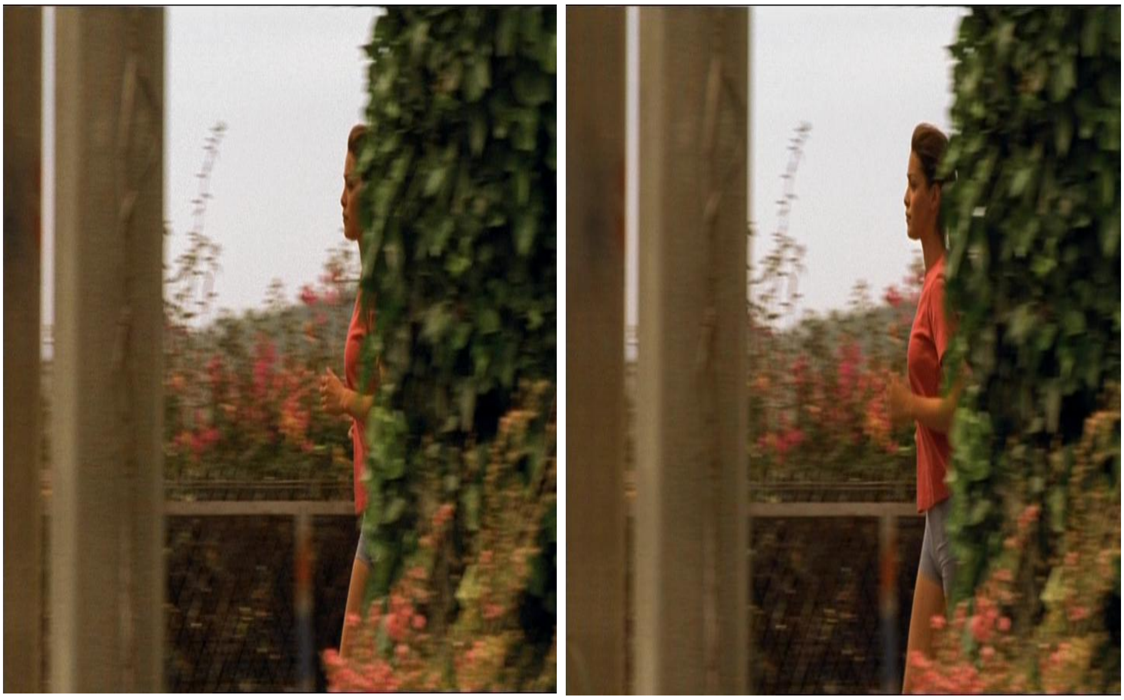


Grad. Sm. +OCC+OBMC
(PSNR:31.472 dB, SSIM:0.969)

Orginal frame 46

Figure 4.9: Table Tennis video sequence results

In the following figures 4.11 and 4.12, we demonstrate the improvements achieved by gradient based smoothing approach. If you focus on the girl's right shoulder, it is covered in previous frame and it is uncovered in the following. In the previous version of the algorithm, there occurs artifacts on her right shoulder. However, in the latest version, we add the gradient error criteria to track the motion of object boundaries correctly. As seen from the figures we achieve better visual quality with the new error criteria, Gradient ABPD.



Girl test sequence frame 32

Girl test sequence frame 33

Figure 4.10: Adjacent test frames in Girl sequence

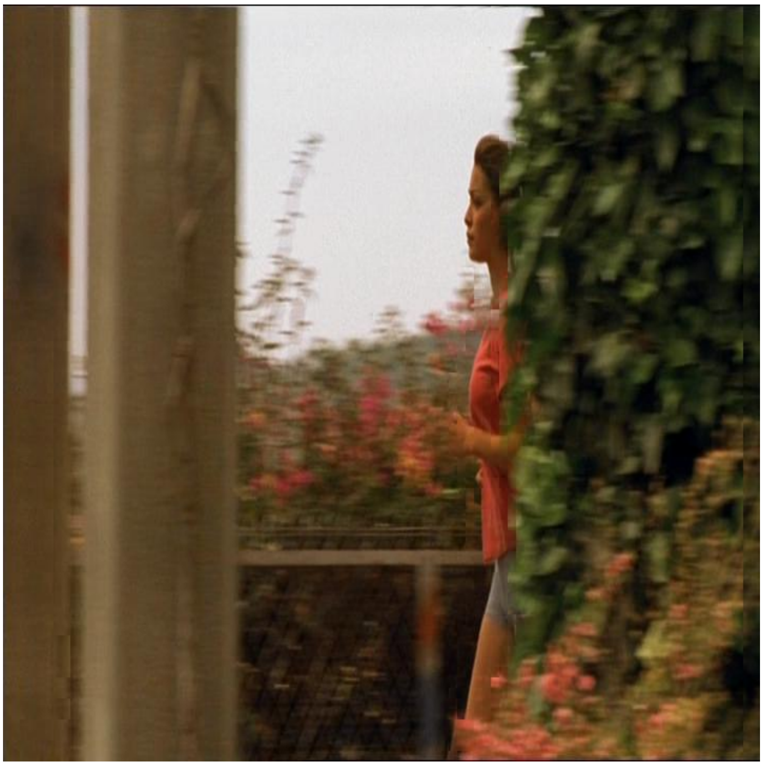


Figure 4.11: OBMC scheme without Gradient ABPD smoothing

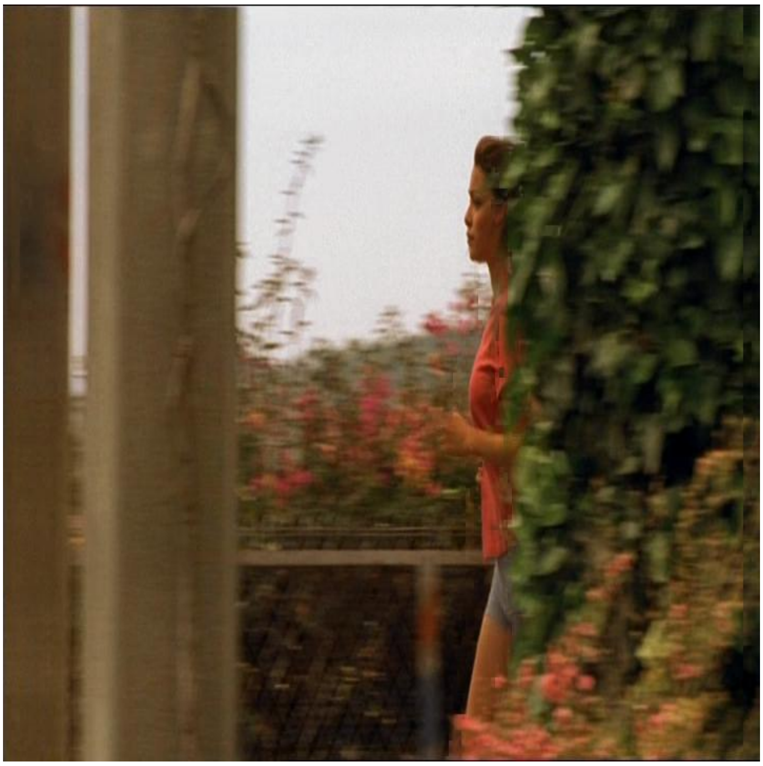


Figure 4.12: OBMC scheme with Gradient ABPD smoothing

Conclusion

In this dissertation, we present a new approach to video frame rate up-conversion problem for HDTV. In chapter 2, a general description of motion estimation techniques is discussed and the needs for true motion field estimator are given. Two important true motion trackers, a multi-stage motion vector post-processing method for video frame rate up-conversion [4] and 3-D recursive search [3] are described in detail. In chapter 3, The algorithms discussed in chapter 2 [4], [3] are used to estimate an initial true motion field and then applied to our video frame rate up-conversion system which has a gradient based motion vector smoothing and occlusion aware overlapped block motion compensated filter for picture rate interpolation. In chapter 4, the subjective and objective measurements are presented in detail and the proposed OBMC scheme is compared with other methods in the literature. All in all, the proposed OBMC scheme has a better performance than the proposed methods [6],[7] and [10].

The performance of the proposed algorithm can be improved with more advanced covered/uncovered decision supported by background/foreground differentiation. Also motion boundary tracking and motion based segmentation techniques can be investigated for better occlusion decisions and correct motion vector assignment in occluded blocks. Frame rate conversion will be an important research area in the upcoming years and together with super-resolution video resolution enhancement still remain an open field for innovative research.

References

[1] A. Murat Tekalp. *Digital Video Processing*. Prentice Hall Signal Processing Series, Prentice Hall PTR Upper Saddle River, NJ 07458, USA, 1995.

[2] Gerard de Haan. *Video Processing for Multimedia Systems*. University Press Eindhoven, Eindhoven, The Netherlands, 2003.

[3] H. Huijgen O.A. Ojo G. de Haan, P.W. Biezen. True motion estimation with 3-d recursive search block matching. In *IEEE Transactions Circuits and Systems Video Technologies*. IEEE Press.

[4] T.Q. Nguyen A. M. Huang. A multi-stage motion vector processing method for motion compensated frame interpolation. *IEEE Transactions on Image Processing*, 18:740–752, 2009.

[5] T.Q. Nguyen A. M. Huang. Correlation-based motion vector processing for motion compensated frame interpolation. *IEEE Transactions on Image Processing*, 17:694–708, 2008.

[6] C. S. Kim S.J. Ko B.D. Choi, J. W. Han. Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation. *IEEE Transactions onCircuits and Systems for Video Technology*, 17:407–416, 2007.

[7] B. Xiao X. Gao, Y. Yang. Adaptive frame rate up-conversion based on motion classification. *Elsevier, International Journal of Signal Processing*, 88.

[8] M. Burl G.A. Thomas. Vector assignment for video image motion compensation. *US Patent, Patent no:6,005,639.*

[9] S. K. Lee Y. Kim S. J. Kang, D. G. Yoo. Multiframe-based bilateral motion estimation with emphasis on stationary caption processing for frame rate up-conversion. *IEEE Transactions on Consumer Electronics*, 54:1830–1838, 2008.

[10] H. Blume. Nonlinear vector error tolerant interpolation of intermediate video images by weighted medians. *Elsevier Signal Processing, Image Communication*, 14.

[11] Y. Paker A. Hamosfakidis. A novel hexagonal search algorithm for fast block matching motion estimation. *EURASIP Journal on Applied Signal Processing*, 2002:595–600, 2002.

[12] Yen-Kuang Chen. *True Motion Estimation-Theory, Application and Implementation.* PhD thesis, Princeton University, November 1998.

[13] M.T. Orchard and G.J. Sullivan. Overlapped block motion compensation: an estimation-theoretic approach. *IEEE Transactions on Image Processing*, 3:693–699, 1994.

[14] H. R. Sheikh E.P Simoncelli Z. Wang, A.C Bovik. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13:600–612, 2004.

[15] H. Marmolin. Subjective mse measures. *IEEE Transactions on Systems, Man and Cybernetics*, 16:486–489, 1986.

[16] T. G. Noll J. von Livonius, H. Blume. Design of a pareto-optimization environment and its application to motion estimation.

[17] J. L. Wu Y. T. Yang, Y. S. Tung. 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, 17:1700–1713, 2007.

[18] C.N. Cordes A. Heinrich and G. de Haan. Improved picture-rate conversion using classification-based lms-filters. In *Electronic Imaging; Visual Communications and Image Processing*, number 7257–56, San Jose, CA, U.S., 2009. IS&T/SPIE.

[19] C. H. Cheung L.M. Po. Novel cross-diamond-hexagonal search algorithms for fast block motion estimation. *IEEE Transactions on Multimedia*, 7:16–22, 2005.

[20] C.N. Cordes A.Heinrich, G. de Haan. A novel performance measure for picture rate conversion methods. In *International Conference on Consumer Electronics, (ICCE 2008)*. IEEE.

[21] J.G.W.M. Janssen J. R. Braspenning R. Wittebrood E.B. Bellers, J.W. van Gurp. Solving occlusion in frame-rate up-conversion. In *International Conference on Consumer Electronics, (ICCE 2007)*. IEEE.

[22] C. Bartels and G. de Haan. Occlusion classifiers for picture rate conversion. In *Electronic Imaging; Visual Communications and Image Processing*, number 7257–49, San Jose, CA, U.S., 2009. IS&T/SPIE.

[23] Y. Liu W. Zhang Y. Ling, J. Wang. Spatial and temporal correlation based frame rate up-conversion. In *International Conference on Image Processing, (ICIP 2008)*, pages 909–912, San Diego, California, U.S., 2008. IEEE.

[24] H.F. Ateş B.Çizmeci. Occlusion artifact removal in video frame rate up-conversion. In *17th Signal Processing and Communications Applications Conference, (SIU 2009)*, pages 277–280, Antalya, Turkey, 2009. IEEE.

[25] Ya-Qin Zhang Yao Wang, Jrn Ostermann. *Video Processing and Communications*. Prentice Hall, 2001.

[26] John W. Woods. *Multidimensional Signal, Image and Video Processing and Coding*. Academic Press, imprint of Elsevier, Renssealer Polytechnic Institute, New York, Troy, USA, 2006.

Curriculum Vitae

Burak Çizmeci was born in 23 May 1985, in Konya. He received his B.S. degree in Electronics Engineering in 2007, B.S. degree in Computer Engineering in 2008 both from Işık University. He worked as a teaching assistant at the department of electronics engineering of Işık University from 2007 to 2009. His research interests include video coding, de-noising, frame rate up-conversion and super-resolution.

Publications

[1] B.Çizmeci, H.F. Ateş, Occlusion artifact removal in video frame rate up-conversion. *In 17th Signal Processing and Communications Applications Conference*, (SIU 2009), pages 277-280, Antalya, Turkey, 2009. IEEE.

[2] E. Tamer,B.Çizmeci, A different approach for license plate recognition system. *In 17th Signal Processing and Communications Applications Conference*, (SIU 2009), pages 357-360, Antalya, Turkey, 2009. IEEE.