VIDEO FRAMES INTERPOLATION USING ADAPTIVE WARPING

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Ying Chen Lou

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

December 2010

Purdue University

West Lafayette, Indiana

ACKNOWLEDGMENTS

I would like to begin by giving thanks to my major advisor, Professor Mark J.T. Smith, for his guidance and support both financially and mentally. His attitudes towards his work and good habits influenced me deeply. As the head of ECE department, he had numerous meetings to attend during the day, however, he managed to focus on his research and other work by always being the first one to go to work, and keeping a block of nondisruptive time to get concentrated. I am sincerely grateful that he gave me opportunities to attend many conferences and freedom to try several internships, which broadened my scope and linked my knowledge from academia to industry.

I would also like to thank my co-advisor, Professor Edward Delp. He provides me the lab equipment and other resources which are essential for my whole PhD program. He is strict on many things, however, his quick grasp of the ideas and pointing out the fundamental problems steered me to the right direction and prevented me from being distracted. I would like to extend my thanks to my doctoral committee members, Professors Mary Comer and Michael D. Zoltowski, for their encouragement, helpful comments, and insight.

I am also thankful for the support and friendship from my colleagues in the Video and Image Processing (VIPER) lab. It has been a really joyful and memorable journey to work with those smart people: Dr. Cuneyt Taskiran, Dr. Hyung Cook Kim, Dr. Zhen Li, Dr. Anthony Martone, Dr. Limin Liu, Dr. Liang Liang, Dr. Nitin Khanna, Ashok Mariappan, Aravind Mikkilineni, Maggie Fengqing Zhu, Satyam Srivastava, Ka Ki Ng, Kein Lorenz, Marc Bosch, Albert Parra, Meilin Yang, Chang Xu, Bin Zhao. I would like to thank other graduate students at Purdue who I met over the years: Zhongqiang Huang, Guangzhi Cao, Zhou Yu, Jianing Wei, Rong Zhang, Wei Zhang, etc. They gave me a lot of help and support in both my academic and personal life.

Finally, I owe my deepest gratitude to my family. I want to thank my parents, Jianbao Chen and Guoyuan Shen, for their unconditional love, support, and prayers. This work

would not be possible without them. I am very fortunate to have known my beloved husband, Willie X. Lou, during my PhD journey, who is also my best friend. His love, patience, and understanding made my life more joyful and meaningful. I could not imagine those rough times without him being there to support me. This thesis is dedicated to my whole family.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

ABBREVIATIONS

HDTV	high definition television
SDTV	standard definition television
PDA	personal digital assistant
DVD	digital versatile disc
GPS	global positioning system
HR	high resolution
LR	low resolution
MRI	magnetic resonance imaging
3D	three-dimentional
2D	two-dimentional
OFE	optical flow equation
EMSE	estimated mean square error
MMSE	minimum mean square error
WSE	weighted square error
CGI	control grid interpolation
ACGI	adaptive control grid interpolation
SVD	singular value decomposition
LM	Levenberg-Marquardt
PSNR	peak signal-to-noise ratio
MPEG	moving picture experts group
DCT	discrete cosine transform
LL	low low
MCP	motion-compensated prediction
GOP	groups of pictures

ABSTRACT

Lou, Ying Chen. Ph.D., Purdue University, December 2010. Video Frames Interpolation Using Adaptive Warping . Major Professors: Mark J.T. Smith and Edward Delp.

In this dissertation, a strategy for high fidelity scaling of video frames to higher resolutions is introduced. The method employed is a combination of block matching-based motion estimation and optical-based motion estimation, which builds on the Control Grid Interpolation (CGI) methods. Low resolution images are interpolated and used to obtain motion vectors that are then used to warp high definition reference frames. Because motion estimation is an ill-posed problem, robustness is critical in the motion search procedure. To improve the robustness of the derived motion fields, a bidirectional motion estimation method is proposed. A hierarchical motion structure is used to solve the ambiguity in the framework.

Improving the spatial quality of video coded at low bit rates is a problem of general interest. Toward this end, a combined forward-backward warping method is proposed to improve the resolution of video encoded with H.264/AVC. The scheme attempts to capture long-term spatial detail by warping high resolution reference frames in accordance with displacement vectors derived from decoded low resolution frames. At low bit rates, the proposed method can achieve better PSNRs and better subjective quality than conventional H.264/AVC for sequences with low to moderate motion.

The third application considered in this dissertation is video frame rate up-conversion. A new inter-frame motion compensated interpolation method is proposed that employs motion vector correction based on residual energy. Experimental results show that it can improve the visual quality of the interpolated frames where competing methods fail.

1. INTRODUCTION

1.1 Motivation and Application

Video and image processing has had a tremendous impact over the years on commercial products, such as cell phones, digital cameras, media players, portable digital versatile disc (DVD) players, and portable global positioning systems (GPS). Among the techniques employed in those digital devices, spatial resolution enlargement and temporal interpolation are of great interest.

Spatial resolution enlargement techniques can be applied to several applications. The first application is the display of standard definition television (SDTV) signals on high definition television (HDTV) screens. Nowadays larger and larger displays appear in the market. The majority of television programming is in SDTV format and cannot be displayed directly on HD displays. Therefore, manufacturers must convert SDTV video to a form compatible with the HD devices, hence the interest in high quality spatial interpolation algorithms.

Another application of spatial enlargement is for camera technology, which is used in cell phones, commercial cameras and computers. In the market, larger and larger pictures are being captured, from 5 Megapixels (MP) to 12 MP, and even higher. To acquire these larger images, large CCD sensor arrays are desirable. However, cost and packaging constraints generally limit the array sizes in portable digital cameras. Thus, using digital signal processing (DSP) algorithms becomes an attractive alternative, particularly because they are cost efficient and upgradable.

In video surveillance, sometimes it is desirable to enlarge a frame for inspection. In many security systems the cameras only record when activated by motion. When something suspicious happens, the investigater reviews the video and will commonly zoom in to see features of interest. Thus spatial enlargement techniques start to play a role.

All the aforementioned applications require spatial interpolation. Many video processing applications involve exploiting properties in the spatio-temporal domain. For many video coding applications, systems seek to operate at reduced rates, in which case temporal resolution is routinely scaled back to better preserve the spatial quality of the sequence. In order to play back the video, temporal interpolation, also called frame-rate up-conversion (FRUC), is performed to restore missing frames. For LCD display applications, high frame rate video is desired in order to reduce blurring, particularly for fast motion video. Quality is improved by up-converting the frame rate of standard video captured at 30 Hz by a factor of two or more. For media broadcast of movies, frame-rate up-conversion is critical to accommodate the frame rate difference of the industry standards. The movie industry typically operates with a 24 frame/second capture rate, while media broadcasts employ a 30 Hz standard. Indeed, there are many applications in which frame-rate up-conversion is necessary and high quality is important.

Resolution enhancement algorithms also play a role in medical imaging. For instance, in magnetic resonance imaging (MRI), there is often a need to reconstruct frames between two slices captured by the system so that the doctors can see an organ more clearly. In other situations, organ movement needs to be captured at high spatial resolution so that a better diagnosis can be performed. The problem is challenging because systems can often only capture high resolution frames at a slow rate and the practical length of timing patients in the MRI machine is very limited. One possible solution is the application developed in this thesis where a high resolution frame is captured periodically and the remaining frames are recorded at low spatial and temporal resolution. In reconstruction, post-processing algorithms involving spatial enlargement and temporal interpolation can be applied to obtain a high definition temporal and spatial recording.

Spatial enlargement and temporal interpolation algorithms can be applied in the video compression domain as well. In the approach developed in this thesis, sequences are downsampled, pre-processed, and transmitted over the network so that less bandwidth is consumed. At the receiver side, the bitstreams are decoded and then post-processing techniques are applied to spatially enlarge the video seqences for display.

Indeed, many applications require either spatial interpolation or temporal interpolation. This has motivated the general framework presented in this thesis, which can be used for both.

1.2 Overview of Previous Work

The general area of image enlargement has a rich history and includes superresolution, frame demosaicing, and frame temporal interpolation. Numerous methods have been reported previously for frame enlargement, the simplest of which is pixel replication [1], perhaps followed by bilinear, bicubic, B-spline and related kernel-based interpolation methods [2], [3]. A number of authors have investigated more intricate approaches aimed at higher quality, such as the method taken by Konto [4] and the method reported by Atkins [5]. Unlike the classical interpolation kernels mentioned above, Konto and Atkins use off-line training to obtain the interpolation filter coefficients. These methods are similar in that images are categorized into classes with each class having its own interpolation filter designed by training. The methods differ in the way classification is handled: Konto [4] uses adaptive dynamic range coding; Atkins [5] uses expectation maximization. Another noteworthy approach was pioneered by Xin et al. [6] who exploit the geometric duality between the high resolution and low resolution image covariances under the assumption that the edge orientations remain the same across scales. This approach does not require training and hence is more convenient to use. These methods are well regarded for enlarging still images. However, greater improvement can be obtained for video sequences by exploiting the inherent temporal information in addition to the spatial information.

A number of papers in the recent literature have used the term "superreolution" in the context of spatial-temporal interpolaiton. These authors have used superresolution to refer to the process of obtaining a high resolution (HR) image or a sequence of HR images from a set of low resolution (LR) observations. The commonly used observation model is shown in Figure 1.1. A continuous scene is first sampled to form a digital HR image f_k. Then it is either warped and blurred or blurred and warped. Following this, downsampling is

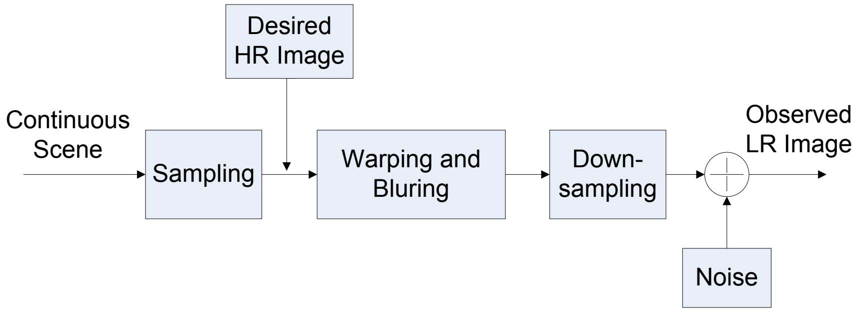


Fig. 1.1. Observation model relating LR images to HR images

performed and noise is added to obtain the observed LR images. The warping process is often translational, or rotational, but can be of higher parametric order. The blurring process often consists of optical blur, and motion blur. The noise can be acquisition noise, registration noise, or quantization noise, or it can be a combination of the three. Most SR methods are distinguished by the type of the reconstruction method used, which observation model is assumed, which domain the algorithm is implemented in (space or freqency), and the assumptions associated with how the LR images are captured.

A good example of exploiting spatio-temporal information is the method reported by Tsai and Huang [7]. They proposed a frequency domain method to detect global translational motion for the purpose of reconstructing a band-limited image from a set of undersampled observation images. This approach was extended by Kim [8] and Bose [9] by incorporating an additive noise model along with a convolution operator to model dispersion. Farsiu et al. [10] proposed an iterative approach based on a bilateral filter using the L1-norm both for the regularization and the data fusion terms in the cost function to deal with different data and noise models.

Maximum a posteriori (MAP) super-resolution methods have also been considered that employ a Bayesian framework to reconstruct the high resolution images. For examples, Tom and Katsaggerlos [11] simultaneously estimate the high-resolution pixel shift, noise variance and super-resolution (SR) image in the MAP model by using Expectation Maximization. Gunturk et al. [12] looked at enlarging reconstructed frames in compressed video, where a MAP framework is employed and quantization parameters are used along with other statistical information. Segall et al. [13] also considered compressed video but in addition addressed blocking artifacts in the reconstruction and smoothing of the edges through manipulating the coding errors within the MAP framework.

Projection onto convex sets (POCS) is an alternative iterative approach that incorporates prior knowledge. Tekalp et al. [14] reported using a validity map and a segmentation map in the POCS framework for robust reconstruction. Patti and Altunbasak [15] reported using a higher order interpolation method within the POCS scheme modifying the constraint set to reduce the ringing artifact in the vicinity of edges. Even though POCS is simple to implement and it can take into account the spatial domain observation model incorporating a prior information, it does not provide a unique solution, is slow to converge, and has a high computational cost. An in-depth survey of such approaches can be found in [16], [17], [18], [19].

Motion estimation has been shown to be important in the registration stages of these methods. Block matching algorithms for motion estimation [20–25] are commonly used in coding and other applications because of their relative simplicity. However, they generally only handle translational motion and thus when rotation or other more complicated motion occurs, block matching algorithms can produce incorrect motion vectors. Optical flow methods are another alternative [26]. Instead of assigning one motion vector to each block, optical flow methods assign to each pixel a specific motion vector, and thus can generate high definition motion fields. However, such methods are often very time consuming and computationally intensive. Optical flow methods also tend to produce erroneous motion vectors if the optical flow estimation is not robust.

1.3 Problem Statement

There are two problems to be addressed here. One is to spatially enlarge the video frames and the other is to temporally interpolate frames. To spatially enlarge video frames, consider a framework involving several low resolution video frames along with one high resolution frame. In a broad sense, the proposed approach can be viewed as a superresolution method because it obtains a sequence of high resolution images from a set of low resolution observations. The small difference is that here additional limited high frequency information is available that can be fused to get more accurate enlarged frames. The primary application motivating this work is lecture video for distance learning where the backgrounds in the video sequences are typically similar. In cases where the lecture video is very long, the number of reference frames can be reduced to a small number and potentially recycled, which makes the framework attractive. For temporal interpolation, two cases are considered. In the first case, the motion vectors are obtained using our proposed framework. In the second case, the motion vectors are computed and transmitted by a codec.

1.4 Outline of the Document

In this document, a general framework is provided which can handle both spatial interpolation and temporal interpolation for video sequences. The core for this framework is a motion estimation algorithm that is a combination of block matching-based motion estimation and optical flow-based motion estimation. The framework builds on the Control Grid Interpolation (CGI) methods in [27–30] and will be introduced in Chapter 2.

The proposed spatial interpolation methods are described in Chapter 3. Spatio-temporal properties of the sequence are exploited in a two-stage algorithm. By having two stages, which consist of purely spatial interpolation and image warping, the enlargement of the "talking head" video sequence is tackled in a way that produces better visual quality.

In Chapter 4, the proposed method is extended to be bidirectional. Because motion estimation is an ill-posed problem, robustness is critical in the motion search procedure. To improve the robustness of the derived motion fields, a bidirectional motion estimation method is used. A hierarchical motion structure is used to resolve ambiguities in the framework.

In Chapter 5, video compression is considered in the context of MPEG standard formats. A new method is proposed and evaluated. Experimental results are presented.

In Chapter 6, temporal interpolation is considered and applied to both uncompressed and compressed video sequences. The problem of performing reconstruction when occlusions are present is discussed. It is shown that these situations can be handled within the proposed framework.

In Chapter 7, the research contributions are summarized and future work is suggested.

2. ADAPTIVE CONTROL GRID INTERPOLATION

In this chapter, optical flow-based and block-based motion estimation are briefly discussed. Adaptive control grid interpolation (CGI) is then introduced. Some of the key issues associated with CGI are connectivity, the error function, and the optimization procedures to estimate the parameters. A quadtree structure is employed in this work and is described in detail.

2.1 Problem Statement

Images can be viewed as projections of three-dimentional objects onto a two-dimentional plane. The intensity of a digital image is denoted $I[n_1, n_2, k]$ where $[n_1, n_2]$ is the spatial location of a given pixel within the image and k denotes the time index. If we assume that for a given sequence the brightness constraints holds, i.e., the luminance of each point remains fixed as it moves, the task of the motion estimator is to find the motion displacement field (d_1, d_2) such that:

$$I[n_1, n_2, k] = I[n_1 + d_1[n_1, n_2, k], n_2 + d_2[n_1, n_2, k], k + \delta k], \tag{2.1}$$

where d_1, d_2 are dependent on the location and time index and are not necessarily integers.

The model described above involves two unknowns and one equation. Because of the underconstrained nature of the formulation, there are several candidates that can lead to different observations of the same scene, most notably illumination variations, certain object movements, and noise effects. For example, consider that we observe image A. If the illumination is increased uniformly that would correspond to adding a value to A resulting in another image B. Alternatively, consider shifting some pixels horizontally and some pixels vertically. We observe 3 different images but there is no motion in the real object. This illustrates one challenge in motion estimation.

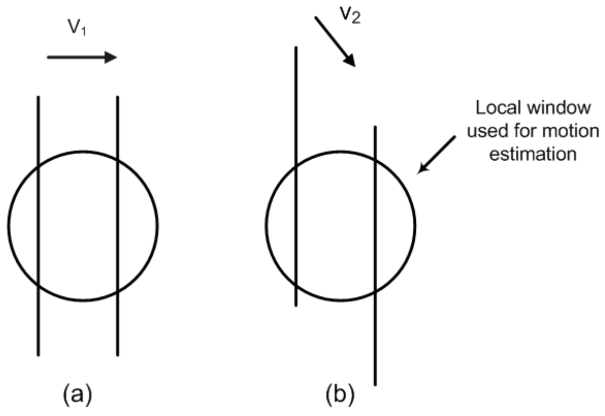


Fig. 2.1. Illustration of the aperture problem.

Another obstacle in any motion estimation algorithm is the aperture problem. The aperture problem refers to the uncertainty of the motion by local observation alone. Figure 2.1 illustrates this case for a binary image in which one line is moving in two different directions. In Figure 2.1(a), the line is moving horizontally and in Figure 2.1(b) it is moving in a diagonal direction where the horizontal speed is the same as the vertical speed. The aperture is smaller than the length of the line and the motion is viewed through the aperture. When viewing through this aperture, we can not distinguish the difference between these two different cases if we focus on one point on the line. So if the aperture size is small, it's impossible to determine the motion without additional information. To determine the motion field, we need to make assumptions about its structure. This can be done by either applying smoothness constraints to the motion fields or by explicitly specifying a motion model.

Motion estimation is an important but difficult problem that has been studied extensively. There are many algorithms discussed in the literature. Distinguishing features among these algorithms are mainly based on two components: the motion model and the estimation process. Two classes of motion models are popular. The first is block-based motion estimation, which assumes a constant motion vector for a block of pixels. The other is pixel-based, which assigns a motion vector to each pixel and uses the optical flow equation and boundary smoothness constraints to derive the motion vectors. The estimation process can use a gradient dependent optimization strategy or a well-defined search strategy.

In the next two sections, we will give a brief overview of those two basic motion estimation algorithms, namely block-based motion estimation and optical flow-based motion estimation.

2.2 Block-Based Motion Estimation

Block-based motion estimation assigns one motion vector to each block. Instead of having an individual motion vector for each pixel, each block shares its motion vectors. Block-based motion estimation distinguishes itself from optical flow-based motion estimation by having its own motion model and by using a different motion search strategy. The motion model used in the algorithm assumes that the pixels within each block undergo the same movement, an assumption only valid for simple translational motion. To estimate the motion parameters, some motion search strategies such as “block matching” are used. The motion vectors are determined by minimizing an error function, such as the mean squared error. The block size and search range play very important roles in getting accurate motion vectors.

2.3 Optical Flow-Based Motion Estimation

In contrast to block-based motion estimation, optical flow-based motion estimation assigns a unique motion vector to each pixel. Thus, it can handle more complex motion. However, it requires more parameters than the block-based method, leading to a more complex motion field. As mentioned before, the optical flow-based motion estimation uses the optical flow equation and additional smoothness constraints to derive the motion fields.

2.3.1 Optical Flow Equation (OFE)

The optical flow equation relates the temporal and spatial derivatives of a point in the video sequence to the instantaneous velocity at that point. The intensity of a pixel at time t, is denoted $I(x(t), y(t), t)$. Taking its derivative with respect to t gives

$$\frac{dI(x(t), y(t), t)}{dt} = \frac{\partial I(x(t), y(t), t)}{\partial x} v_x + \frac{\partial I(x(t), y(t), t)}{\partial y} v_y + \frac{\partial I(x(t), y(t), t)}{\partial t}, \tag{2.2}$$

where $v_x = \frac{dx(t)}{dt}$ and $v_y = \frac{dy(t)}{dt}$ are the horizontal and vertical components of the instantaneous velocities, respectively. If we impose the brightness constraint that the luminance of each point is fixed as it moves, we obtain the canstraint equation

$$\frac{dI(x^*(t), y^*(t)}{dt} = 0, \tag{2.3}$$

where $(x^*(t), y^*(t))$ is the true path of each point. Combining the above two equations leads to the OFE (Optical Flow Equation):

$$\frac{\partial I(x(t), y(t), t)}{\partial x} v_x^* + \frac{\partial I(x(t), y(t), t)}{\partial y} v_y^* + \frac{\partial I(x(t), y(t), t)}{\partial t} = 0. \tag{2.4}$$

The above OFE cannot be solved alone, because there are only one equation and two unknowns. This is related to aperture problem we mentioned before. Additional constraints are needed to find a unique solution.

Equation (2.4) illustrates the relationship between the instantaneous velocity and the partial derivatives at a specific spatial-temporal location, while Equation (2.1) describes the motion displacement from two discrete times k and $k + \delta k$. Using the approximations $v_x \approx \frac{d_1(x,y)}{\delta k}$, $v_y \approx \frac{d_2(x,y)}{\delta k}$, and $\frac{\partial I(x,y,t)}{\partial t} \approx \frac{I(x,y,k) - I(x,y,k+\delta k)}{\delta k}$, we obtain

$$\frac{\partial I(x, y, k)}{\partial x} d_1(x, y) + \frac{\partial I(x, y, k)}{\partial y} d_2(x, y) - I(x, y, k + \delta k) + I(x, y, k) = 0. \tag{2.5}$$

2.3.2 Boundary Smoothness Constraints

The OFE alone cannot give a unique solution. Additional constraints need to be imposed on it. Those constraints are called boundary smoothness constraints or neighborhood constraints and they require the motion field to vary smoothly across the image. The constraints combined with the error functions used to minimize the error lead to different conditions. Some commonly used conditions are discussed below.

Global Smoothness Constriants

This is one of the classical techniques used for OFE [26]. The motion displacement field is supposed to vary smoothly for natural images. An error function exploiting this feature is given by

$$E_p(d_1(n_1,n_2),d_2(n_1,n_2)) + \alpha^2 E_s(d_1(n_1,n_2),d_2(n_1,n_2)) \times E_p(d_1(n_1,n_2),d_2(n_1,n_2)), \tag{2.6}$$

where

$$E_p(d_1,d_2) = \int\int (I_0(\vec{n}) - I_1(\vec{n}) - \frac{\partial I_1(\vec{n})}{\partial n_1} d_1(\vec{n}) - \frac{\partial I_1(\vec{n})}{\partial n_2} d_2(\vec{n}))^2 dn_1 dn_2 \tag{2.7}$$

is the predicted luminance error, $I_0(\vec{n}) = I(\vec{n},k)$, $I_1(\vec{n}) = I(\vec{n},k+\delta k)$, and

$$E_s(d_1,d_2) = \int\int (\frac{\partial d_1(\vec{n})}{\partial n_1})^2 + (\frac{\partial d_1(\vec{n})}{\partial n_2})^2 + (\frac{\partial d_2(\vec{n})}{\partial n_1})^2 + (\frac{\partial d_2(\vec{n})}{\partial n_2})^2 dn_1 dn_2 \tag{2.8}$$

is a smoothness constraint. To balance between minimizing the predicted luminance error and satisfying the smoothness constraint, the parameter α is used. The calculus of variations is used in the optimization procedure, resulting in a Gauss-Seidel iteration to estimate the motion displacement field.

Fixed displacement over a region of support R

This method is also called Wiener-based displacement estimation [31] [32] [33]. It assumes that Equation (2.1) holds for every pixel in region R, which is equivalent to the block translational motion model. By minimizing the estimated mean square error (EMSE), which is

$$\frac{1}{N}\sum_{\vec{n}\in R}(I[\vec{n},k+\delta k] - I[\vec{n},k] - d_1\frac{\partial I[\vec{n},k+\delta k]}{\partial n_1} - d_2\frac{\partial I[\vec{n}],k+\delta k}{\partial n_2})^2, \tag{2.9}$$

we can get a simple solution:

$$\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = A^{-1} \begin{bmatrix} \sum_{\vec{n} \in R} (I[\vec{n}, k+\delta k] - I[\vec{n}, k]) \frac{\partial I[\vec{n}, k+\delta k]}{\partial n_1} \\ \sum_{\vec{n} \in R} (I[\vec{n}, k+\delta k] - I[\vec{n}, k]) \frac{\partial I[\vec{n}, k+\delta k]}{\partial n_2} \end{bmatrix}, \tag{2.10}$$

where

$$A = \begin{bmatrix} \sum_{\vec{n} \in R} (\frac{\partial I[\vec{n}, k+\delta k]}{\partial n_1})^2 & \sum_{\vec{n} \in R} (\frac{\partial I[\vec{n}, k+\delta k]}{\partial n_1})(\frac{\partial I[\vec{n}, k+\delta k]}{\partial n_2}) \\ \sum_{\vec{n} \in R} (\frac{\partial I[\vec{n}, k+\delta k]}{\partial n_1})(\frac{\partial I[\vec{n}, k+\delta k]}{\partial n_2}) & \sum_{\vec{n} \in R} (\frac{\partial I[\vec{n}, k+\delta k]}{\partial n_2})^2 \end{bmatrix}. \tag{2.11}$$

Equation (2.10) is applied repeatedly until a suitable stopping condition is satisfied.

Directional Smoothness Constraints

This constraint requires the motion displacement field to vary smoothly along the boundaries instead of across boundaries [34] [35]. It use second order statistics to impose the global smoothness constraint in the direction perpendicular to the image gradient. It is enforced by adding a weight matrix to the norm used in Equation (2.8).

Edge Constraints

The disadvantage of the global smoothness constraint is it tends to blur the motion field boundaries. To overcome this disadvantage, the smoothness constraint near the edge boundaries can be relaxed [36] [37].

Hierarchical Constraints

This method applies the smoothness constraints across the scales [38]. The scale of an image is defined as the size or resolution of the image. A penalty term is used to reduce the chance that motion vectors change significantly from one scale to the next.

2.3.3 Lucas-Kanade Approach and Its Limitation

Lucas-Kanade (LK) optical flow-based motion estimation [39] was originally proposed to solve the image registration problem for stereo video systems. The problem can be formulated as given functions $F(x)$ and $G(x)$ which should be a shifted version of $F(x)$, what is the displacement h that can lead to the minimum difference between $F(x+h)$ and $G(x)$. Using Eq. (2.4) is not enough because it has only one equation but two unknowns. LK approached the problem by assuming that (1) pixels within a small region R undergo the same motion; (2) the motion is very small thus the first order approximation Taylor expansion can be used on pixel value approximation, i.e., $F(x+h) = F(x) + hF'(x)$.

To minimize the least square error between $F(x+h)$ and $G(x)$, the motion movement h is derived as $h \doteq \frac{\sum_x F'(x)[G(x)-F(x)]}{\sum_x F'(x)^2}$. In the 2D domain, the equation becomes

$$h = [\sum_x (\frac{\partial F}{\partial x})^T [G(x) - F(x)]][\sum_x (\frac{\partial F}{\partial x})^T (\frac{\partial F}{\partial x})]^{-1}, \tag{2.12}$$

where $\frac{\partial}{\partial x}$ is the gradient operator with respect to x, as a column vector.

Compared to numerous other optical flow computation methods such as Horn-Schunk [26], Fleet-Jepson [40], Black and Amandan [41], Nagel [42], Proesmans [43], Uras et al [44] , Lucas-Kanade method is considered to be relatively more accurate, robust, and faster. However, in addition to the fundamental limitations of optical-based methods, because the nature of Lucas-Kanade method is local based, it is easily affected by noise, which is common among differential-based optical flow methods. Moreover, the motion field produced by LK is sparser than other methods, which might be a problem for some applications.

2.4 Motivation

Block-based motion estimation has a compact representation of motion vectors and has a significant but relatively light computational requirement. Typically, simple search strategies are used under the assumption of a translational motion model. However, if higher orders motion models are used, the computational complexity increases significantly. The translational motion model cannot represent true motion and thus often contributes to visual distortion. Optical flow-based motion estimation can assign individual motion vectors to each pixel value and thus can describe all types of motion. It has more flexibility but it is computationally more intensive and the motion field is quite complex.

The limitations on traditional methods motivate the development of a method that combines features of both the block-based motion estimation and the optical flow-based motion estimation to take advantage of both techniques. By having a hybrid motion model, we gain flexibility and a compact representation. The approach we consider here is block-based, because we assume the motion model over each block. However, within each block, we use the optical-flow equation to optimize the motion parameters.

2.5 Control Grid Interpolation (CGI) Model Definition

The general model CGI has been explored by several authors [27] [28] before. The basic model we discuss here builds on the work of Monaco [29] and Frakes [30]. It assumes the motion displacement field $[d_1, d_2]$ within each region R is a linear combination of the basis functions, which are dependent on locations.

Displacements are denoted by

$$d_1(\vec{n}) = \sum_{i=1}^{p} \alpha_i \theta_i(\vec{n}) \tag{2.13}$$

$$d_2(\vec{n}) = \sum_{i=1}^{q} \beta_i \phi_i(\vec{n}) \tag{2.14}$$

where $\theta_i(\vec{n})$ and $\phi_i(\vec{n})$ are independent basis functions used to describe the motion fields and are dependent on the location $\vec{n}$. The parameters α_i and β_i must be estimated. We can also rewrite the above equations using vector notation

$$d_1(\vec{n}) = \vec{\alpha}^T \vec{\theta}(\vec{n}) \tag{2.15}$$

$$d_2(\vec{n}) = \vec{\beta}^T \vec{\phi}(\vec{n}) \tag{2.16}$$

where $\vec{\alpha} = [\alpha_i \ldots \alpha_p]^T$, $\vec{\beta} = [\beta_i \ldots \beta_q]^T$, $\vec{\theta}(\vec{n}) = [\theta_i(\vec{n}) \ldots\ \theta_p(\vec{n})]^T$ and $\vec{\phi}(\vec{n}) = [\phi_i(\vec{n}) \ldots \phi_q(\vec{n})]^T$.

As pointed out by Monaco [29], this model has more flexibility because the basis functions can be written in many ways, enabling us to represent different models.

1 The translational model, which equates to block-based motion estimation, is obtained when
p=q=1,
$\theta_1(\vec{n}) = \phi_1(\vec{n}) = 1.$

2 The affine model is obtained when
p=q=3,
$\theta_1(\vec{n}) = \phi_1(\vec{n}) = 1,$
$\theta_2(\vec{n}) = \phi_2(\vec{n}) = n_1,$
$\theta_3(\vec{n}) = \phi_3(\vec{n}) = n_2.$

3 The bilinear model is obtained when
p=q=4,
$\theta_1(\vec{n}) = \phi_1(\vec{n}) = 1,$
$\theta_2(\vec{n}) = \phi_2(\vec{n}) = n_1,$
$\theta_3(\vec{n}) = \phi_3(\vec{n}) = n_2$
$\theta_4(\vec{n}) = \phi_4(\vec{n}) = n_1 n_2.$

Other models can also be represented, however we don't list them here. In the extreme case, if the number of basis functions is the same as the pixel number in the region R, it can represent the optical flow-based motion field. All of this is to say that the general model has the flexibility of representing a lot of motion models including all those related to this work.

In our algorithm, we use a model order with $p = q = 4$ and basis functions

$$\theta_1(\vec{n}) = \phi_1(\vec{n}) = (\tfrac{n_1^2 - n_1}{n_1^2 - n_1^1})(\tfrac{n_2^2 - n_2}{n_2^2 - n_2^1}), \tag{2.17}$$

$$\theta_2(\vec{n}) = \phi_2(\vec{n}) = (\tfrac{n_1^2 - n_1}{n_1^2 - n_1^1})(\tfrac{n_2 - n_2^1}{n_2^2 - n_2^1}) \tag{2.18}$$

$$\theta_3(\vec{n}) = \phi_3(\vec{n}) = (\tfrac{n_1 - n_1^1}{n_1^2 - n_1^1})(\tfrac{n_2 - n_2^1}{n_2^2 - n_2^1}) \tag{2.19}$$

$$\theta_4(\vec{n}) = \phi_4(\vec{n}) = (\tfrac{n_1 - n_1^1}{n_1^2 - n_1^1})(\tfrac{n_2^2 - n_2}{n_2^2 - n_2}) \tag{2.20}$$

where (n_1^1, n_2^1) and (n_1^2, n_2^2) denote the coordinates of the upper left corner and the bottom right corner of a rectangular region R respectively. We show the example of a rectangular region in Figure 2.5 labelled with this coordinate system.

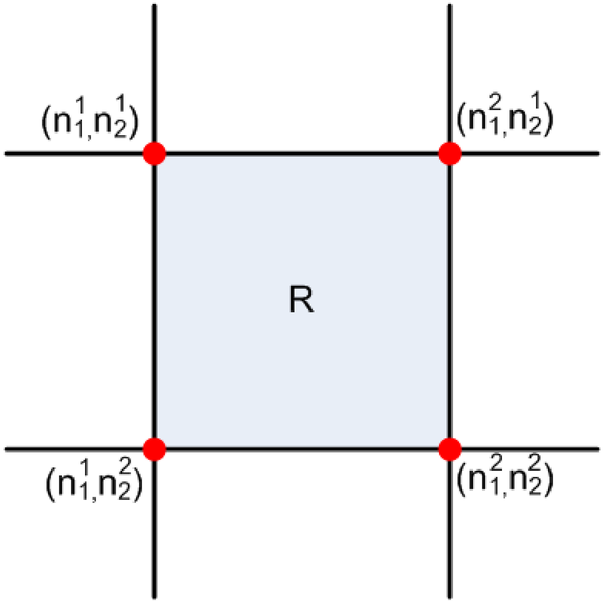


Fig. 2.2. Coordinates of region R in CGI

When (n_1, n_2) is at the corner, only one of the four basis functions is equal to one, while the others have a value of zero.

2.6 Connectivity

The CGI can be characterized as connected, disconnected or adaptive. The connected model, shown in Figure 2.6(a), requires continuity along the edges between two adjacent regions. The disconnected model, illustrated in Figure 2.6(b), allows discontinuity at the boundaries of each region. The adaptive model, which is a combination of the two, allows only some discontinuity at the boundaries.

For the connected model, one corner will affect its surrounding regions and is not totally independent. To estimate the motion parameters, it is usually more computationally intensive than disconnected model. However it has the advantage of being able to represent

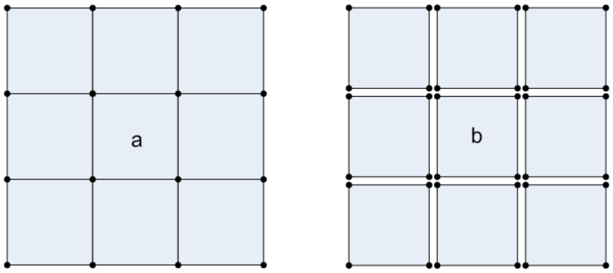


Fig. 2.3. Illustration of connectivity. (a) Connected model. (b) Disconnected model.

the motion more compactly. The disconnected model has different motion parameters for each region and the boundary areas don't affect their surrounding neighbors. Thus, optimization procedures used to estimate the motion parameters are independent and require fewer iterations, which makes it less computational intensive. However, it has more motion parameters to estimate. To illustrate this point, we can take the Figure 2.6 for example. In Figure 2.6(a), we only need totally 16 independent motion vectors to estimate but in Figure 2.6(b) we will have 36 independent motion vectors to estimate. Thus, the disconnected model is less compact than connected model. The adaptive model takes advantage of both models and is expected to perform best.

2.7 Taylor Expansion and Error Function

The motion displacement fields d_1, d_2 at each pixel are a linear combination of the basis functions. To estimate $p + q$ motion parameters from Equation 2.13 and 2.14, we need to add additional smoothness conditions as mentioned in Section 2.3.1. In our case, we need to estimate a total of eight parameters. Since the original optical flow equation is for continuous signals, the optical flow equation at time $k + \delta k$ is expanded using the Taylor series expansion to solve for discrete signal and to make the iteration process easier.

The original equation 2.1

$$I[\vec{n}, k] = I[n_1 + d_1[n_1, n_2, k], n_2 + d_2[n_1, n_2, k], k + \delta k] \tag{2.21}$$

is then converted to

$$I[\vec{n},k] \approx I[\vec{n},k+\delta k] + \frac{\partial I[\vec{n},k+\delta k]}{\partial n_1} d_1(\vec{n}) + \frac{\partial I[\vec{n},k+\delta k]}{\partial n_2} d_2(\vec{n}) \tag{2.22}$$

where $\vec{n} = (n_1, n_2)$. Substituting Equations 2.15 and 2.16 into Equation 2.22, we get

$$I[\vec{n},k] \approx I[\vec{n},k+\delta k] + \frac{\partial I[\vec{n},k+\delta k]}{\partial n_1} \vec{\alpha}^T \vec{\theta}(\vec{n}) + \frac{\partial I[\vec{n},k+\delta k]}{\partial n_2} \vec{\beta}^T \vec{\phi}(\vec{n}) \tag{2.23}$$

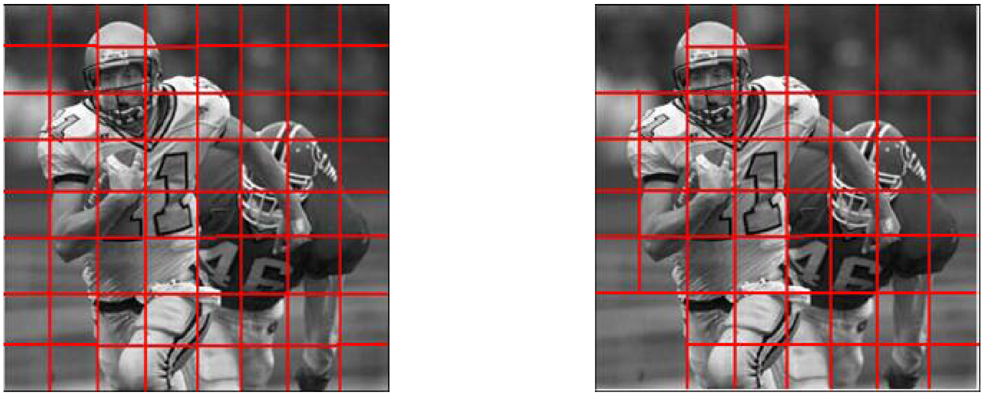
Equation 2.23 is supposed to be valid for all the pixels within region R. To calculate the motion displacement fields, the weighted square error function (WSE)

$$E(\vec{\alpha},\vec{\beta}) = \sum_{\vec{n}\in R} W(\vec{n}) p(I[\vec{n},k] - I[\vec{n},k+\delta k] - \frac{\partial I[\vec{n},k+\delta k]}{\partial n_1} \vec{\alpha}^T \vec{\theta}(\vec{n}) - \frac{\partial I[\vec{n},k+\delta k]}{\partial n_2} \vec{\beta}^T \vec{\phi}(\vec{n})) \tag{2.24}$$

is minimized where $W(\vec{n})$ is a weighting function. $p(.)$ is a function which can be chosen for the specific application. For example, if it is quadratic, it is equivalent to MSE. By choosing different weighting functions, it can be targeted for different applications. If there is no displacement to be emphasized or deemphasized, the weighting function can be uniform. Using the smoothness constraints mentioned in Section 2.3.1 we can minimize Equation 2.24 to avoid an incoherent displacement field. The optimization method will be described in the section 2.9.

2.8 Qualdtree Structure and Adaptive Splitting

The CGI algorithm is modified to be adaptive to the content of the image to alleviate the computational intensity. Since it is a combination of both the block-based motion estimation and the optical flow-based motion estimation, when the block size is large, for some images, it can not capture the complex nature of the motion field within the block when the image is divided into even size blocks. To overcome this shortcoming, an adaptive mechanism is needed. As the high value of WSE indicates the situation when the block is not small enough to well capture the motion field within the region, we can keep splitting the region R using some quadtree structure. Figure 2.4 can be used to illustrate this situation. Figure 2.4(a) uses uniform subdivision to estimate the motion parameters within each block while Figure 2.4(b) uses an adaptive splitting mechanism to determine whether the block needs to be further divided. As we can see that in areas where the content is quite simple we can use larger block size to capture the motion field while in areas where there are more details, a smaller block size is used.



(a) (b)

Fig. 2.4. (a) Uniform subdivision algorithm (b) Adaptive splitting using quadtree structure

The algorithm starts with the evenly divided block size partitioning of the image. When the WSE exceeds a predefined threshold, the blocks are divided into smaller blocks using quadtree structure. If the WSE is small enough, there is no need to further split the block and those motion parameters can represent the motion field well within that region. If the WSE is still large, this algorithm keeps dividing the blocks until either the WSE is below a predefined threshold or a minimum block size is reached. This modified Control Grid Interpolation method is called Adaptive Control Grid Interpolation (ACGI).

2.9 Optimization Procedure

To derive the motion parameters for each block, we need to use Equation (2.24). It is well known that the quadratic error function is sensitive to outliers [45]. Outliers occur when complex motion is present in the same region or when uncovered background or occlusion occurs. To make it more robust, we can consider other more robust functions

[46] [27] at the cost of being more computationally intensive. For simplicity, we use the quadratic error function. So the approximate error function becomes:

$$E(\vec{\alpha},\vec{\beta}) = \sum_{\vec{n}\in R} W(\vec{n})(I[\vec{n},k]-I[\vec{n},k+\delta k]-\frac{\partial I[\vec{n},k+\delta k]}{\partial n_1}\vec{\alpha}^T\vec{\theta}(\vec{n})-\frac{\partial I[\vec{n},k+\delta k]}{\partial n_2}\vec{\beta}^T\vec{\phi}(\vec{n}))^2 \tag{2.25}$$

The parameters can be expressed as

$$\begin{bmatrix}\vec{\alpha}\\ \vec{\beta}\end{bmatrix} = A^{-1}\begin{bmatrix}\sum_{\vec{n}\in R}(I[\vec{n},k+\delta k]-I[\vec{n},k])\frac{\partial I[\vec{n},k+\delta k]}{\partial n_1}\vec{\theta}(\vec{n})\\ \sum_{\vec{n}\in R}(I[\vec{n},k+\delta k]-I[\vec{n},k])\frac{\partial I[\vec{n},k+\delta k]}{\partial n_2}\vec{\phi}(\vec{n})\end{bmatrix}, \tag{2.26}$$

where

$$A = \begin{bmatrix}\sum_{\vec{n}\in R}(\frac{\partial I[\vec{n},k+\delta k]}{\partial n_1})^2 & \sum_{\vec{n}\in R}(\frac{\partial I[\vec{n},k+\delta k]}{\partial n_1})(\frac{\partial I[\vec{n},k+\delta k]}{\partial n_2})\\ \sum_{\vec{n}\in R}(\frac{\partial I[\vec{n},k+\delta k]}{\partial n_1})(\frac{\partial I[\vec{n},k+\delta k]}{\partial n_2}) & \sum_{\vec{n}\in R}(\frac{\partial I[\vec{n},k+\delta k]}{\partial n_2})^2\end{bmatrix}. \tag{2.27}$$

The choice of the weighting function for a robust estimation can be found in [27]. The procedure of minimizing the WSE function is an iterative process which can be summarized as the following steps:

1 Initialization

Initialize the parameters $\vec{\alpha}$ and $\vec{\beta}$ and set the iteration counter $s = 0$. Evaluate the true error function with the initial values and store the result.

2 Taylor Series Expansion

Perform Taylor series expansion of $I[n_1+\vec{\alpha}_s^T\vec{\theta}(\vec{n}), n_2+\vec{\beta}_s^T\vec{\phi}(\vec{n}), k+\delta k], \vec{n} = (n_1, n_2) \in R$

3 Parameter estimation and Update

Apply optimization (for example, using the gradient descent algorithm) to derive $\vec{\alpha}$ and $\vec{\beta}$ by minimizing the approximate WSE error function for each of the control points for a given region in succession, while holding the other three fixed. More about this process is discussed later.

4 Evaluation

Evaluate the WSE in Eq (2.25) and store the parameter error if the error function decreases.

5 Termination Test

Evaluate the stopping condition. If convergence is not achieved, go to step 2 and increase the iteration counter s. The convergence is discussed in the next section.

The Taylor series expansion is performed at $(n_1 + d_1(\vec{n}), n_2 + d_2(\vec{n}))$ in step 2 to obtain a more accurate approximation as the error function in Eq (2.25) is just an approximation to simplify the optimization process. There is no guarantee that minimizing the approximate error function actually minimizes the true error function. In general, the algorithm tends to reduce both the approximate error and true error by large amounts in the first several iterations. Most of the results obtained by minimizing the approximate error coincide with minimizing the true error.

In step 3, an optimization algorithm is used. This part will be explained in the following.

2.9.1 Matrix Inverse Update

The easiest way to get the parameters is to invert the matrix given in Eq (2.27). However, direct calculation of the inverse of A is often not possible as A might be singular or near singular. The chance of A being singular increases when the support region R is small or when a large number of basis functions are used for a particular region. Also if the assumption of the linear variation in the luminance is always true and the motion in the scene can be precisely described by this model, then A will often be singular.

To avoid the singularity problem, a singular value decomposition method (SVD) [45] is introduced. Instead of computing the inverse of A directly, we first find the SVD, which decomposes A into $A = USV^T$. S is a diagonal matrix consisting of singular values and U and V are unitary so that $A^{-1} = VS^{-1}U^T$. When some elements of S are zero or close to zero, the pseudoinverse can be obtained by replacing those elements with zero. Thus, if A is singular, the SVD gives an update vector whose elements are minimal in the mean squared error sense [45].

2.9.2 Levenberg-Marquardt Update

Instead of computing the inverse matrix, we can use a gradient descent algorithm directly on Eq (2.25).

Taking the derivative of Eq (2.25) with respect to the model parameters gives:

$$\frac{\partial E(\vec{\alpha},\vec{\beta})}{\partial \vec{\alpha}} = \sum_{\vec{n}\in R} 2W(\vec{n})(I[\vec{n},k] - I[n_1 + \vec{\alpha}^T\vec{\theta}(\vec{n}), n_2 + \vec{\beta}^T\vec{\phi}(\vec{n}), k+\delta k])\vec{\theta}(\vec{n}) \tag{2.28}$$

$$\frac{\partial E(\vec{\alpha},\vec{\beta})}{\partial \vec{\beta}} = \sum_{\vec{n}\in R} 2W(\vec{n})(I[\vec{n},k] - I[n_1 + \vec{\alpha}^T\vec{\theta}(\vec{n}), n_2 + \vec{\beta}^T\vec{\phi}(\vec{n}), k+\delta k])\vec{\phi}(\vec{n}) \tag{2.29}$$

The gradient descent algorithm gives

$$\vec{\alpha}^{n+1} = \vec{\alpha}^{n} + \frac{1}{\lambda}\frac{\partial E(\vec{\alpha}^n,\vec{\beta}^n)}{\partial \vec{\alpha}} \tag{2.30}$$

$$\vec{\beta}^{n+1} = \vec{\beta}^{n} + \frac{1}{\lambda}\frac{\partial E(\vec{\alpha}^n,\vec{\beta}^n)}{\partial \vec{\beta}} \tag{2.31}$$

The Levenberg-Marquardt method is a combination of the gradient descent and the inverse-Hessian method used to update non-linear objective functions [45]. In areas where the EMSE is a poor approximation, the gradient descent algorithm is used, while in other cases, the inverse-Hessian matrix is used. The equation governing the Levenberg-Marquardt method is

$$(A+\lambda I)\begin{bmatrix}\vec{\alpha}\\ \vec{\beta}\end{bmatrix} = \vec{g}, \tag{2.32}$$

where $\vec{g}$ is the gradient of Eq (2.25). When $\lambda \gg 1$ the Levenberg-Marquardt method is equal to Eqs (2.30) and (2.31). When $\lambda \ll 1$, it reduces to Eq (2.26).

To incorporate the LM algorithm into the optimization, we can break step 3 into three parts.

1 Solve Eq. (2.32) using SVD to find $\vec{\alpha}^{n+1}$ and $\vec{\beta}^{n+1}$

2 Evaluate the WSE in Eq. (2.25)

3 If the error decreases, divide λ by 10 and continue. Otherwise, multiply λ by 10 and continue to step 1 unless the maximum number of iterations has been reached.

The three parts that demand large computation in the optimization algorithm in decreasing order are calculating matrix A, finding gradients for the Taylor series expansion, and evaluating the error function. Because in the LM, we don't need to recalculate A and update the Taylor series expansion, adding the loop above in the LM doesn't increase complexity.

2.10 Convergence and Initial Conditions

Two important issues generally associated with optimization are convergence and initial starting points. One challenge in the optimization problem is how to avoid local minimum. Unfortunately, since the error surface of the function in Eq. (2.25) has numerous local minima, convergence to the global minimum can't be guaranteed without resorting to more complex global optimization algorithms. Usually the algorithm will converge rapidly toward a minimum in the first few iterations. After then the incremental improvement tends to decrease.

The starting points are also very crucial as they determine whether or not the global minimum is reached and how many iterations will be needed for convergence. Our major concern when choosing the starting points is to avoid being trapped in the local minima. To address this we use a hierarchical estimation strategy where the starting points of the inner blocks are the last stored value of the parameters for the outer block. For the same initial partitioning block, the subsequent starting points are the stored parameters of the previous neighboring block.

3. VIDEO FRAME SPATIAL INTERPOLATION FOR UNCOMPRESSED VIDEO SEQUENCE

In this chapter, we incorporate the motion models described in Chapter 2 to address the problem of video frame spatial enlargement. Instead of conventional interpolation methods such as bilinear, bicubic, and spline, which only consider spatial information, we also take into account temporal information. The overall system is presented first, followed by a detailed description of the inner workings of the constituent components. The three main components are spatial interpolation, warping, and post processing. Three variations of the algorithm are considered: the first uses full-band warping; the second uses high frequency band warping; the third uses bidirectional warping with a hierarchical motion structure. The first two methods are explained in detail in this chapter. The third is briefly introduced as a prelude to the full description in Chapter 4.

3.1 Problem Statement

In the past decades, many facets of video frame interpolation have been explored. In particular, consideration has been given to spatial interpolation, temporal interpolation, and spatio-temporal interpolation. Spatial interpolation has the longest history. With the emerging popularity of digital video, research in temporal interpolation has become more popular, where the goal is to reconstruct new frames between two existing frames. A common approach to temporal interpolation is to perform motion estimation and then use the motion vectors to synthesize the intermediate frames. The challenge typically is how to estimate these motion vectors accurately. Recognizing that statistical dependencies are present in both space and time. Spatio-temporal interpolation algorithms have also been explored. Instead of using spatial information alone to do frame resolution enhancement, additional temporal information in the sequence is used to make the enlarged frames look sharper.

The algorithms introduced in this thesis are spatio-temporal in nature, involving a mix of high and low frame rates and resolutions. More specifically, the video sequences include a sequence of a high resolution frame followed by several low resolution frames. Mixed resolution scenarios are not uncommon. For instance in medical imaging, there are application where the doctors need to watch the video capturing the patient's organ movement in order to diagnose a particular condition. However, capturing even one high resolution frame is very time consuming. To make it worse, the patient needs to stay still for a long time which can be problematic. To shorten the video capturing time, one solution is to periodically capture high resolution frames while the frames in between are captured at low resolution. Later on, post-processing methods can be applied to those low resolution frames offline. Then the doctors can use those spatially processed enlarged video to diagnose the patients. Another example is related to consumer cameras that have a video capture mode . Memory limitations discourage direct acquisition of high resolution video. But taking high resolution still pictures is common and economical. Thus, the available information is a mix of high resolution and low resolution imagery, which could be processed to obtain high resolution video. In lecture video scenarios, due to the limited motion, high resolution frames and low resolution frames share much similarity. If the videos are captured at low resolution along with reference frames captured periodically at high resolution, the memory saving can be very large. The algorithm developed as part of this thesis operate in this mixed resolution environment.

The crucial part of the problem is how to fuse the additional information from the high resolution video frame into the low resolution video frames. Without using the high resolution frame, the interpolated video frames lack high frequency detail. In the next section, we provide the overall system diagram that describes our approach.

3.2 Notation and System Diagram

The task at hand is to interpolate $N \times M$ video frames, which we denote X_k (where k denotes the frame number) to frames of size $2N \times 2M$, denoted as Y_k. In addition to having the $N \times M$ frames X_k, we assume we have a reference frame A, which is a high quality $2N \times 2M$ frame from the video. One of the most straightforward approaches to enlarge X_k is to upsample the image and perform interpolation to obtain Y_k. There are several classical ways to spatially interpolate the frames, such as bilinear, bicubic, and spline interpolation, which are discussed in Section 3.3. As our gold standard, we use bilinear interpolation in our comparisons.

The overall system diagram associated with our method can be parsed into three subsystems: spatial interpolation, warping, and post processing as illustrated in Figure 3.1.

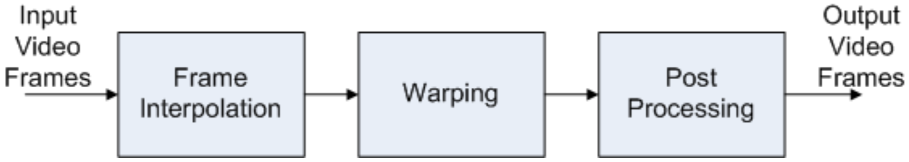


Fig. 3.1. System diagram

In the spatial interpolation sub-system, we first upsample the input and apply bilinear interpolation as described in Section 3.3. In the next sub-section, we apply a dense field motion estimation algorithm (discussed in Chapter 2) between the reference frame A and the interpolated frame Y_k. Frame A, which has very high spatial resolution, is then warped into frame k in one of two ways described in Section 3.4. The warped frame is then used as the interpolated output, which we denote Z_k. In the final sub-system, called post processing, we correct those motion vectors that are not reliable.

An important part of the success of this approach is to be able to perform the dense field motion vector estimation accurately and efficiently and to perform high quality warping. In this work we use a modified adaptive control grid approach [29] as discussed earlier.

At the onset, each frame is partitioned into small contiguous blocks. Within each block, we assume that the intensity of pixels remains unchanged but with some degree of motion displacement. Thus the pixel displacements can be computed by the optical flow equation.

3.3 Spatial Interpolation

Many method have been developed over the years for spatial image interpolation. Some of them are simple, some are complex. In this section, we provide an overview of these interpolation methods to provide context for the new algorithms we introduce later in the thesis.

3.3.1 Classic Interpolation Methods

The classical interpolations methods are very well known, and are described here briefly since they are referenced and used later on for benchmarking. They employ an interpolation kernel, the order of which distinguishes one method from another. The simplest method is nearest neighbor interpolation, also known as zero-order interpolation, where interpolated pixels take on the value of the nearest neighbor. This method is attractive because of its simplicity, but suffers from jagged edge effects along object boundaries. The next method in the progression is first-order interpolation, illustrated pictorially in Figure 3.3.1. Given pixel values at integer position, pixel values at non-integer positions (n_1, n_2) can be calculated using the equation (3.1).

$$I(n_1, n_2) = h_2^T \begin{bmatrix} I(\lfloor n_1 \rfloor, \lfloor n_2 \rfloor) & I(\lceil n_1 \rceil, \lfloor n_2 \rfloor) \\ I(\lfloor n_1 \rfloor, \lceil n_2 \rceil) & I(\lceil n_1 \rceil, \lceil n_2 \rceil) \end{bmatrix} h_1 \tag{3.1}$$

where

$$h_1 = \begin{bmatrix} h(n_1 - \lfloor n_1 \rfloor) \\ h(n_1 - \lceil n_1 \rceil) \end{bmatrix}, \quad h_2 = \begin{bmatrix} h(n_2 - \lfloor n_2 \rfloor) \\ h(n_2 - \lceil n_2 \rceil) \end{bmatrix}, \tag{3.2}$$

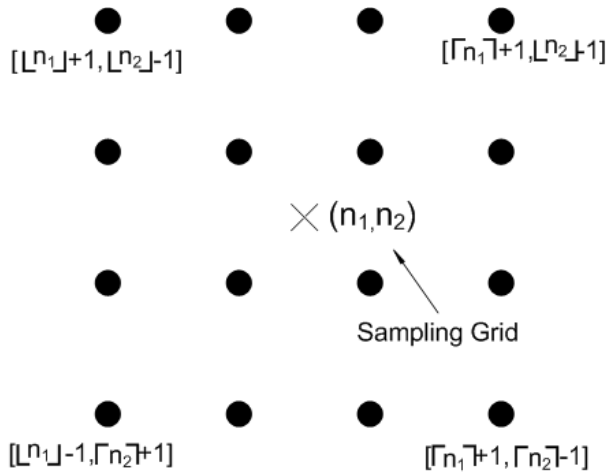

$[\lfloor n_1 \rfloor+1, \lfloor n_2 \rfloor-1]$
$[\lceil n_1 \rceil+1, \lfloor n_2 \rfloor-1]$
$\times\ (n_1, n_2)$
Sampling Grid
$[\lfloor n_1 \rfloor-1, \lceil n_2 \rceil+1]$
$[\lceil n_1 \rceil+1, \lceil n_2 \rceil-1]$

Fig. 3.2. Illustration of non-integer pixel interpolation using neighboring pixels.

and

$$h(n) = 1 - |n|.$$

Bicubic spline interpolation is similar to bilinear interpolation in the sense of using neighboring pixels. The difference is that instead of using a 2 × 2 neighborhood, it uses a 4 × 4 neighborhood so that the kernel is a better approximation to the ideal lowpass function. The governing equation is given by

$$I(n_1, n_2) = h_2^T T h_1 \tag{3.3}$$

where

$$h_1 = \begin{bmatrix} h(n_1 - \lfloor n_1 \rfloor + 1) \\ h(n_1 - \lfloor n_1 \rfloor) \\ h(n_1 - \lceil n_1 \rceil) \\ h(n_1 - \lceil n_1 \rceil - 1) \end{bmatrix}, \tag{3.4}$$

$$h_2 = \begin{bmatrix} h(n_2 - \lfloor n_2 \rfloor + 1) \\ h(n_2 - \lfloor n_2 \rfloor) \\ h(n_2 - \lceil n_2 \rceil) \\ h(n_2 - \lceil n_2 \rceil - 1) \end{bmatrix}, \tag{3.5}$$

$$T = \begin{bmatrix} I(\lfloor n_1 \rfloor - 1, \lfloor n_2 \rfloor - 1) & I(\lfloor n_1 \rfloor, \lfloor n_2 \rfloor - 1) & I(\lceil n_1 \rceil, \lfloor n_2 \rfloor - 1) & I(\lfloor n_1 \rfloor + 1, \lfloor n_2 \rfloor - 1) \\ I(\lfloor n_1 \rfloor - 1, \lfloor n_2 \rfloor) & I(\lfloor n_1 \rfloor, \lfloor n_2 \rfloor) & I(\lceil n_1 \rceil, \lfloor n_2 \rfloor) & I(\lfloor n_1 \rfloor + 1, \lfloor n_2 \rfloor) \\ I(\lfloor n_1 \rfloor - 1, \lfloor n_2 \rfloor) & I(\lfloor n_1 \rfloor, \lfloor n_2 \rfloor - 1) & I(\lceil n_1 \rceil, \lfloor n_2 \rfloor - 1) & I(\lfloor n_1 \rfloor + 1, \lfloor n_2 \rfloor - 1) \\ I(\lfloor n_1 \rfloor - 1, \lceil n_2 \rceil) & I(\lfloor n_1 \rfloor, \lceil n_2 \rceil) & I(\lceil n_1 \rceil, \lceil n_2 \rceil) & I(\lceil n_1 \rceil + 1, \lceil n_2 \rceil) \\ I(\lfloor n_1 \rfloor - 1, \lceil n_2 \rceil + 1) & I(\lfloor n_1 \rfloor, \lceil n_2 \rceil + 1) & I(\lceil n_1 \rceil, \lceil n_2 \rceil + 1) & I(\lceil n_1 \rceil + 1, \lceil n_2 \rceil + 1) \end{bmatrix} \tag{3.6}$$

and

$$h(n) = \begin{cases} 1.5|n|^3 - 2.5|n|^2 + 1, & 0 \leq |n| < 1 \\ -0.5|n|^3 + 2.5|n|^2 - 4|n| + 2, & 1 \leq |n| < 2 \\ 0 & 2 \leq |n|, \end{cases} \tag{3.7}$$

3.3.2 New Edge-Directed Interpolation by Xin and Orchard

Another image interpolation methods that has become popular is the new edge-directed interpolation method (NEDI) [6]. We use it as a benchmark in the later sections. The major idea in the new directed interpolation method is that there exists a geometric duality in the covariances of the low and high resolution frames. That is to say, pairs of pixels at different resolutions along the same orientation are coupled in a way that can be exploited.

High resolution image pixels in the high resolution image may be estimated from the four neighboring pixels along diagonal directions. Covariance coefficients within a local window from the low-resolution image are estimated first. Then those covariance coefficients are used in the interpolation process for the high-resolution image.

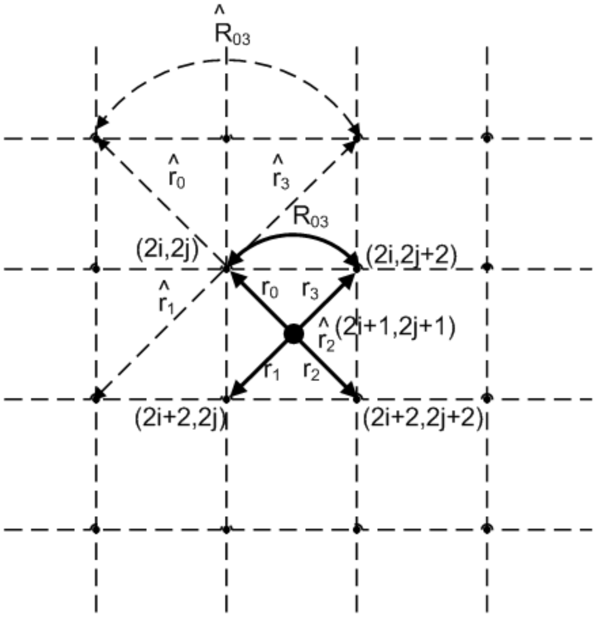


Fig. 3.3. Geometric duality when interpolating $Y_{2i+1,2j+1}$ from $Y_{2i,2j}$.

By modeling the image as a locally stationary Gaussian process, the method can adapt to an arbitrarily oriented edge that has an infinite scale and thus improves the subjective quality of interpolation. Figures 3.3 and 3.4 illustrate the geometric duality between the low resolution image and the high resolution image. Given a low-resolution image $X_{i,j}$ of size N x M, which comes directly from a high-resolution image of size $2N \times 2M$, i.e., $Y_{2i,2j} = X_{i,j}$, the authors use the fourth-order linear interpolation

$$\hat{Y}_{2i+1,2j+1} = \sum_{k=0}^{1} \sum_{k=0}^{1} \alpha_{2k+l} Y_{2(i+k),2(j+l)} \tag{3.8}$$

to estimate the missing pixels $Y_{2i+1,2j+1}$.

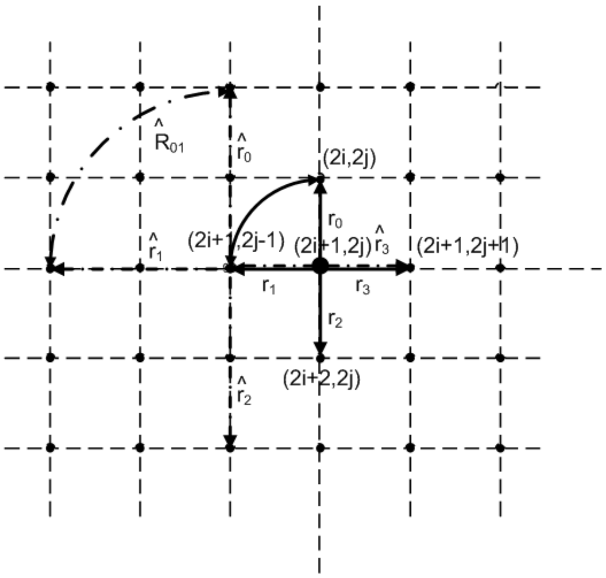


Fig. 3.4. Geometric duality when interpolating $Y_{i,j}(i+j = odd)$ from $Y_{i,j}(i+j=even)$.

Initially all four coefficients α_{2k+l} $(k = 0,1, l = 0,1)$ are unknown. Estimates are obtained from the low resolution image within a local window $K \times K$ according to the geometric duality by using the classical covariance method

$$\hat{R} = \frac{1}{K^2} C^T C, \quad \hat{\vec{r}} = \frac{1}{K^2} C^T \vec{y}, \tag{3.9}$$

where $y = [y_1, y_2, \ldots, y_i, \ldots, y_{K^2}]$ is the data vector listing the $K \times K$ pixels inside the local window and C is a $4K^2$ data matrix whose ith column vector is comprised of the four nearest neighbors of y_i along the diagonal direction.

By classical Wiener filtering theory, the MMSE criterion produces optimal MMSE linear coefficients α given by

$$\begin{aligned} \vec{\alpha} &= R^{-1}\vec{r} && (3.10) \\ &= (C^T C)^{-1}(C^T \vec{y}). && (3.11) \end{aligned}$$

By substitute Eq. (3.11) into Eq. (3.8) we can get pixels $Y_{2i+1,2j+1}$. At this stage, we know all the pixel values $Y_{i,j}$ $(i+j=\text{even})$. But we still need to get $Y_{i,j}$ $(i+j=\text{odd})$.

The authors also point out that Fig. 3.3 and Fig. 3.4 are isomorphic up to a scaling factor of $2^{\frac{1}{2}}$ and a rotation factor of $\frac{\pi}{4}$. So we can get $Y_{i,j}$ $(i + j = \text{odd})$ in the same way by using its four nearest neighbors along the diagonal direction, which we estimate from the previous step.

The method is reported to work well for natural images predominately containing edges with infinite scales. It has been noted that in images with certain types of texture patterns, for example, where edges are tightly packed (so-called finite scale edges), the frequency aliasing from downsampling can corrupt the true edge orientation. The method is also very computational intensive. The authors [6] report that the overall complexity is roughly a couple orders of magnitude higher than that of linear interpolation.

3.3.3 Adaptive Synthesis System for Interpolation

A number of approaches were investigated in this regard, the most interesting of which is the use of adaptive synthesis filter banks [47,48]. This approach was motivated by earlier work [49–51] in which it was shown that adaptive filter banks could be implemented in a subband coder and could be designed to preserve the exact reconstruction property in the absence of quantization. This approach allowed the filters to change in response to the input signal characteristics, which in turn could be used to improve the subjective quality in a subband coder. Similar to this adaptive subband coding scenario, adaptive synthesis filters can also be considered for the interpolation stage of the proposed video coding system. A block diagram of the synthesis system is shown in Figure 3.5.

The synthesis filter bank is adaptive in the sense that it switches among a set of filters on the fly in response to input signal characteristics. If all the synthesis filters are chosen to be the same, then the structure devolves to the classic interpolation structure.

Much of the experimental work performed thus far has employed non-adaptive interpolation filters. However, good results have recently been observed using adaptive filters in which the phase is switched between minimum phase, maximum phase, and linear phase in response to the strength of edges in the input image [47, 48, 51]. Similarly, good results can also be observed by dynamically switching directionality. That is, in regions of the image where the dominant signal variance is oriented horizontally and vertically, classical row-column filters can be used. And, in regions where the signal variance is dominant in diagonal directions, filtering can be performed along diagonal directions. Employing these two adaptive strategies in combination can lead to good interpolation results with sharper subjective quality and/or with less blocking and ringing distortion.

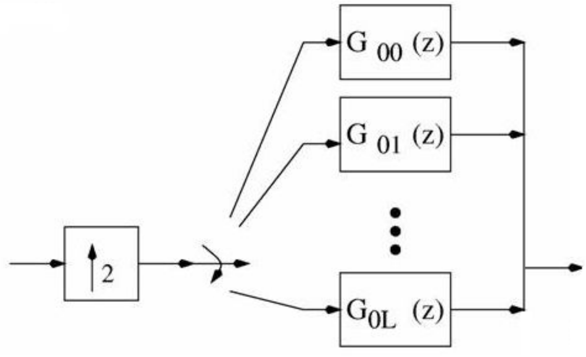


Fig. 3.5. Adaptive synthesis system for interpolation [47, 48]

3.4 Incorporation of ACGI

The basic underpinning of ACGI were discussed in Chapter 2. In this thesis, we expand ACGI to include three variants (a) using full-band warping, (b) high frequency band warping, and (c) bidirectional warping. The full-band warping method simply involves warping the high resolution frame into the interpolated frame via the method mentioned in Chapter 2.

The second variant of the algorithm, the high frequency band warping method, employs only a high frequency version of the reference frame in the warping process. The high resolution frame is denoted as A with resolution $2N \times 2M$. The low resolution frames are denoted as X_k with k indicating the frame number. The interpolated frames for X_k are Y_k. A high spatial frequency reference frame is computed, which we denote $\tilde{A}$. That is, $\tilde{A}$ is a highpass filtered version of A with cutoff frequency $\frac{\pi}{2}$. The dense motion fields between the reference frame and Y_k are then used to warp $\tilde{A}$ to frame k, the result of which we denote D_k. The algorithm output Z_k is then given by $Z_k = D_k + Y_k$.

The third variant considers the fundamental problem of the optical flow method and tries to enhance the robustness by using forward warping and backward warping. Each of these variants is presented in detail in the following sections together with the new interpolation algorithm. Performance results are provided at the end of each section.

3.4.1 Full-band Warping

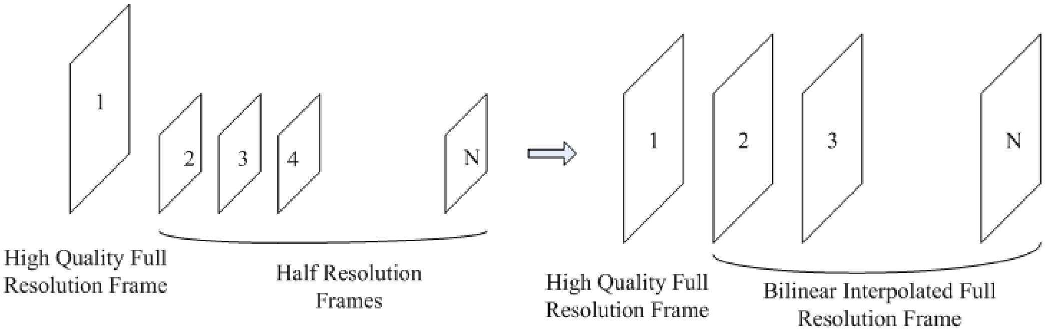

1
2
3
4
N
High Quality Full Resolution Frame
Half Resolution Frames
1
2
3
N
High Quality Full Resolution Frame
Bilinear Interpolated Full Resolution Frame

Fig. 3.6. First stage of full band warping: frame bilinear interpolation

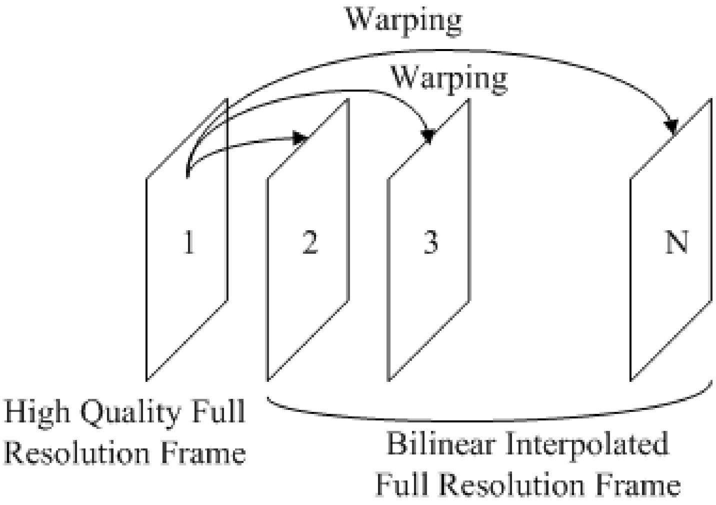

Warping
Warping
1
2
3
N
High Quality Full Resolution Frame
Bilinear Interpolated Full Resolution Frame

Fig. 3.7. Second stage of full band warping

The full-band warping consists of two steps. The first step is to interpolate the low resolution frame into a high resolution frame so that all frames are the same size. Any one of the interpolation methods mentioned previously can be used in this process. Figure 3.6 illustrates this process and gives a pictorial description of the output after this stage. The next step is to warp the high quality frame into different subsequent low quality interpolated frames, as depicted in Figure 3.7.

The general method operates on frame pairs (e.g. X_1 and X_2), like conventional MCP (Motion Compensated Prediction) methods. If X_1 and X_2 are the source frame and target frame respectively, the goal is to warp X_1 into X_2.

Let $X_1[i,j]$ denote the intensity of the pixels in frame X_1 at position $[i,j]$. We assume in this model that pixel intensities remain fixed but move from frame to frame, where the movement can be represented by displacement vectors. At each position

$$X_2[i,j] = X_1[i + d_1[i,j], j + d_2[i,j]], \tag{3.12}$$

where $d_1[i,j]$ is the horizontal motion displacement component at position $[i,j]$ and $d_2[i,j]$ is the vertical motion displacement component. In our implementation , we use the bilinear model discussed in Section 2.6. The displacement vectors are a linear combination of 4 components, which are independent basis functions $\theta[i,j]$.

$$\begin{aligned} d_1[i,j] &= \alpha_1\theta_1[i,j] + \alpha_2\theta_2[i,j] + \alpha_3\theta_3[i,j] + \alpha_4\theta_4[i,j] && (3.13)\\ &= \alpha^T\theta[i,j] && (3.14)\\ d_2[i,j] &= \beta_1\theta_1[i,j] + \beta_2\theta_2[i,j] + \beta_3\theta_3[i,j] + \beta_4\theta_4[i,j] && (3.15)\\ &= \beta^T\phi[i,j]. && (3.16) \end{aligned}$$

In our model the basis functions are associated with a block region R defined by its four corners. Each block then has associated with it a total of 8 parameters to estimate. Frame X_2 is also partitioned into blocks, denoted R. Within each block the mean square error equation is used

$$\sum_{i,j \in R} (X_2[i,j] - X_1[i + \alpha^T\theta[i,j], j + \beta^T\theta[i,j]])^2 \tag{3.17}$$

to estimate the parameters α and β so that the 8 parameters minimize the squared error associated with the block. In our algorithm, Equation (3.17) is approximated by the first two terms of its Taylor series expansion

$$\sum_{i,j \in R} (X_2[i,j] - X_1[i,j] - \frac{\partial X_1[i,j]}{\partial i} \alpha^T \theta[i,j] - \frac{\partial X_1[i,j]}{\partial j} \beta^T \theta[i,j])^2. \tag{3.18}$$

The optimization process uses an iterative gradient-based method in a quad tree framework like the one discussed in Section 2.8. If the error in a block is above a predetermined threshold, we split the block into 4 smaller blocks. This quad tree splitting is repeated until the mean square error threshold criterion is satisfied.

3.4.2 Experimental Results

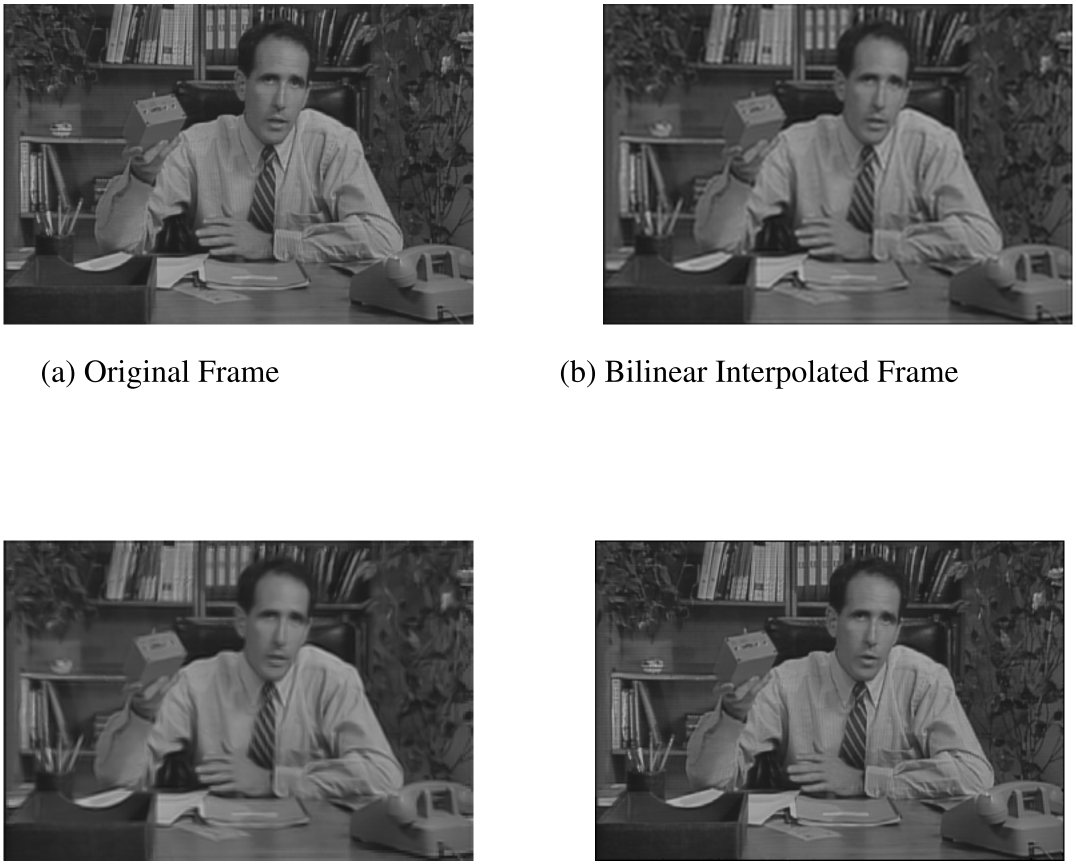
Results from Original Sequences

To assess the performance of our algorithms, we tested them on several “talking head video sequences”: Salesman, Foreman, Akiyo, Carphone, and News. These algorithm were originally designed with “talking head video sequences” in mind. Such video sequences usually have static backgrounds and low motion. All the sequences we tested are CIF (352×288) or SIF (352×240). The mixed spatio-temporal sequence format was obtained by keeping the first frame of the sequences, which serves as the high resolution reference, and downsampled subsequent frames by a factor of two in each direction, the latter serving as our $N \times M$ sequence.

To obtain a better low resolution video frame, we first filter the CIF size frames using a 21-tap half-band lowpass filter. Odd length decimation filtering was chosen to avoid a phase shift in the frequency domain. The order of the filter was chosen to be sufficiently high to mitigate aliasing.

Using the reference frame from the CIF sequence, our aim is to interpolate the 176×144 decimated sequence back to CIF size as just discussed in Section 3.4.1. Here we show some subjective results comparing our results against bilinear interpolated results and results using NEDI [6]. The bilinear interpolated frames are used as our baseline for comparison.

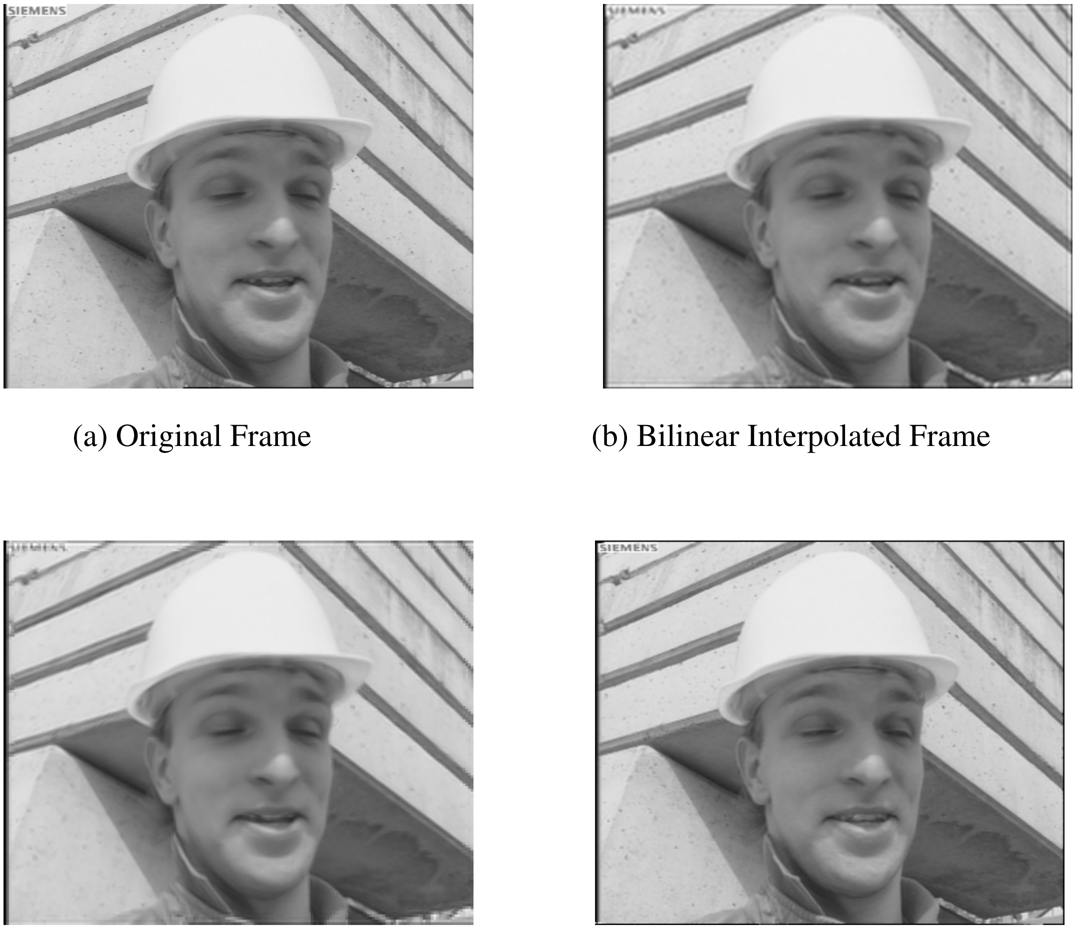
That is, we enlarged the 176×144 decimated sequence to CIF size using classical bilinear interpolation. Figures 3.8 - 3.12 show the subjective results. As we can see, for "talking head" video sequences, the visual improvement is significant. Both bilinear interpolation and NEDI produce blurry results while the full band warping method preserves the detail, which is evident in the bookshelf area, the tie, and the left hand. NEDI achieves better results than the bilinear interpolation. However, the high frequency components are still missing. The full-band warping method keeps the high frequency detail, thus leads to sharper results.



(a) Original Frame (b) Bilinear Interpolated Frame

(c) Frame Interpolated Using NEDI (d) Frame Enlarged Using Full Band Warping

Fig. 3.8. Comparison of the subjective quality of frames from the Salesman sequence.



(a) Original Frame

(b) Bilinear Interpolated Frame

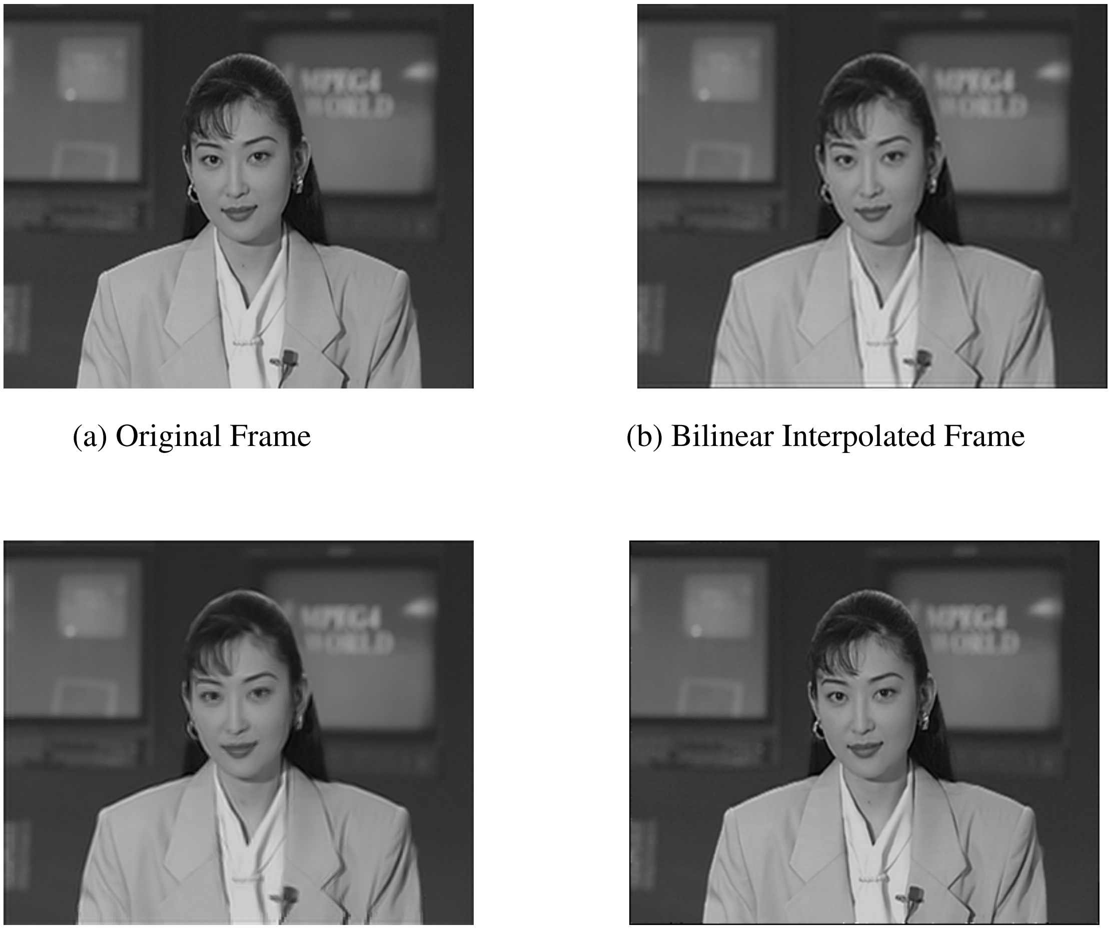
(c) Frame Interpolated Using NEDI

(d) Frame Enlarged Using Full Band Warping

Fig. 3.9. Comparison of the subjective quality of frames from the Foreman sequence.

Influence of Noise on the Robustness of the Algorithm

Often images or videos are corrupted during the capturing, transmitting and receiving stages. The resulting random variations of brightness or color information in images constitute noise. Noise can appear in a number of forms, such as: Gaussian noise, salt-and-pepper noise, shot noise, film grain, and quantization noise, most of which are generated in the capturing process.



(a) Original Frame

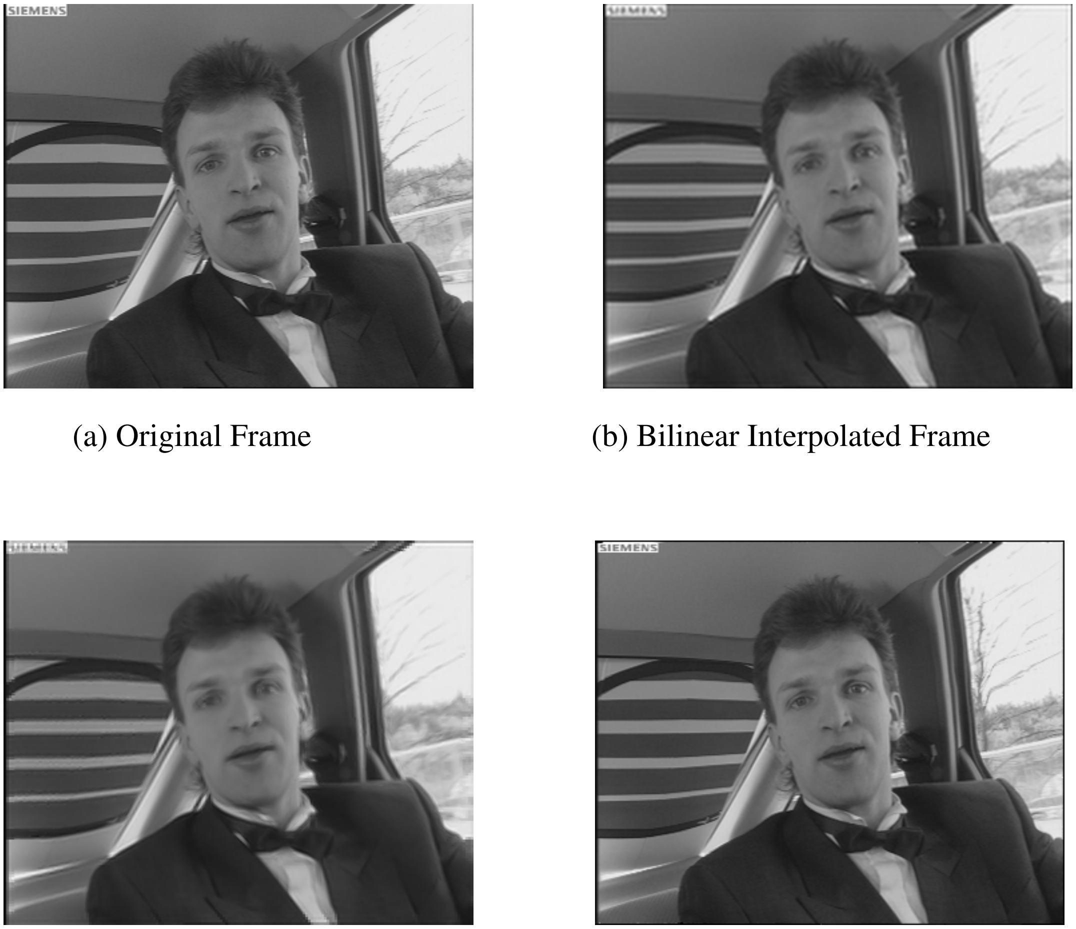
(b) Bilinear Interpolated Frame

(c) Frame Interpolated Using NEDI

(d) Frame Enlarged Using Full Band Warping

Fig. 3.10. Comparison of the subjective quality of frames from the Akiyo sequence.

To test whether an algorithm is robust to noise, Gaussian noise is often used as a test condition. In our experiments, Gaussian noise were added with zero mean and different standard deviations to the “talking head” video sequences. The standard deviations we used were 2, 5, and 10, reflecting different levels of noise; the larger the standard deviation, the lower the quality of the videos. The test video sequences are Akiyo, Foreman, Carphone, News and Salesman. The noise is added to both the reference frame and the low resolution frames with the same standard deviation. Every 5th frame is sampled as the high resolution reference frame.



(a) Original Frame

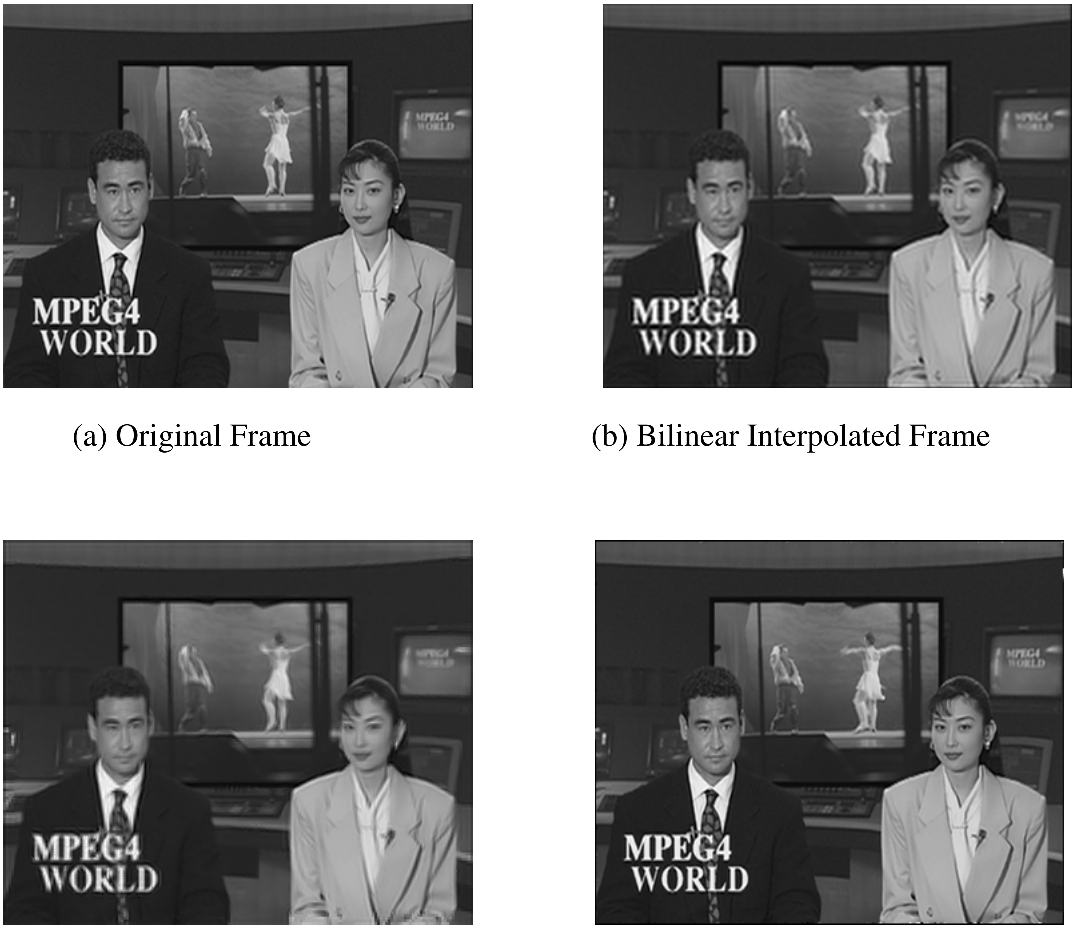
(b) Bilinear Interpolated Frame

(c) Frame Interpolated Using NEDI

(d) Frame Enlarged Using Full Band Warping

Fig. 3.11. Comparison of the subjective quality of frames from the Carphone sequence.

Figures 3.13 - 3.15 compare the full band warping method with bilinear interpolation and NEDI on the noisy Akiyo sequence. In each figure, we can see that the full band warping results achieved better subjective quality. The observation is similar to the results obtained on the original sequences, which were presented in the previous section. The bilinear interpolation results are the worst among the three. NEDI stands in the middle. The full band warping method produces the best visual results because it preserves the detail. Figure 3.13(d), Figure 3.14(d), and Figure 3.15(d) illustrate how the warping method is affected by the noise. As the noise becomes more severe, the quality of interpolated video



(a) Original Frame

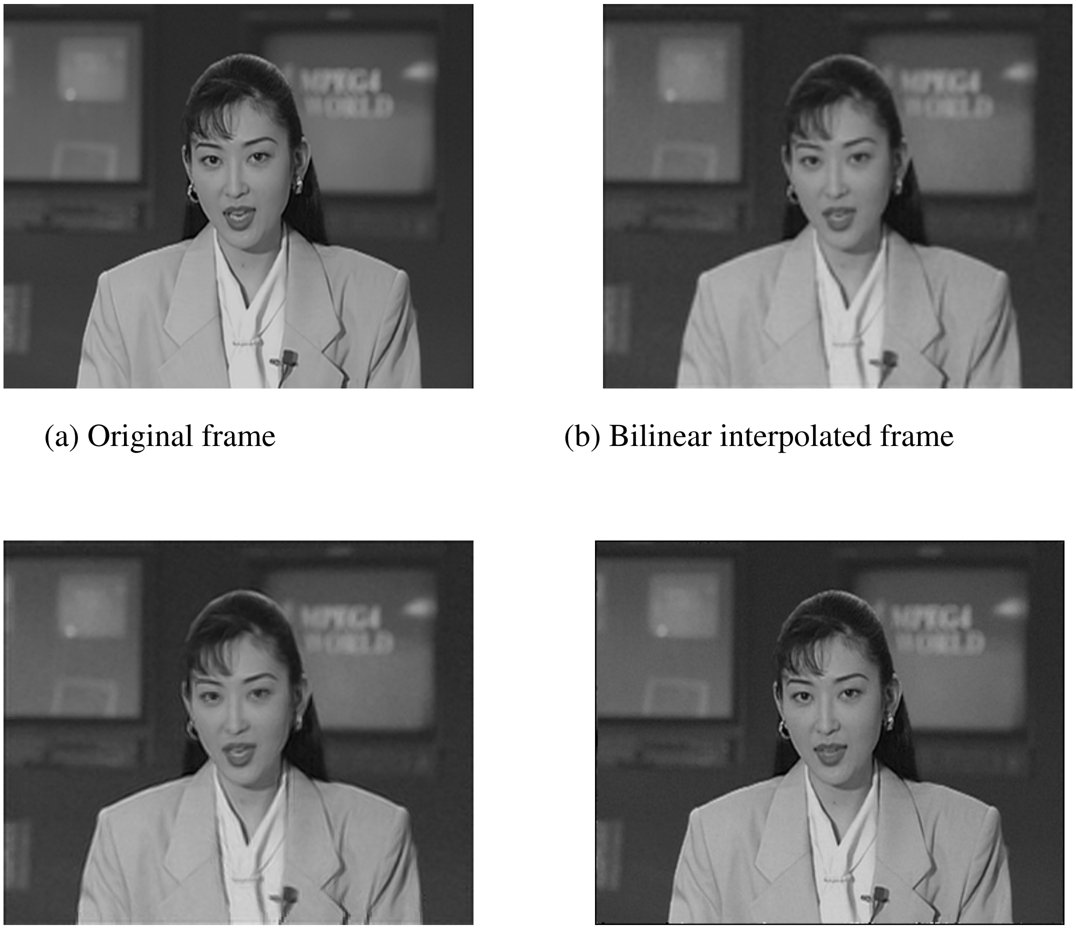
(b) Bilinear Interpolated Frame

(c) Frame Interpolated Using NEDI

(d) Frame Enlarged Using Full Band Warping

Fig. 3.12. Comparison of the subjective quality of frames from the News sequence.

degrades. In Figure 3.13(d), because the added noise is of standard deviation of 2, which is not much, the resulting warped frame still looks very close to the original noisy-free frame. When the standard deviation is increased to 10, the resulting warped frame still maintain its sharpness as shown in Figure 3.15(d), even though the quality is low. In general, additional noise does not affect the sharpness of the warped frame.



(a) Original frame

(b) Bilinear interpolated frame

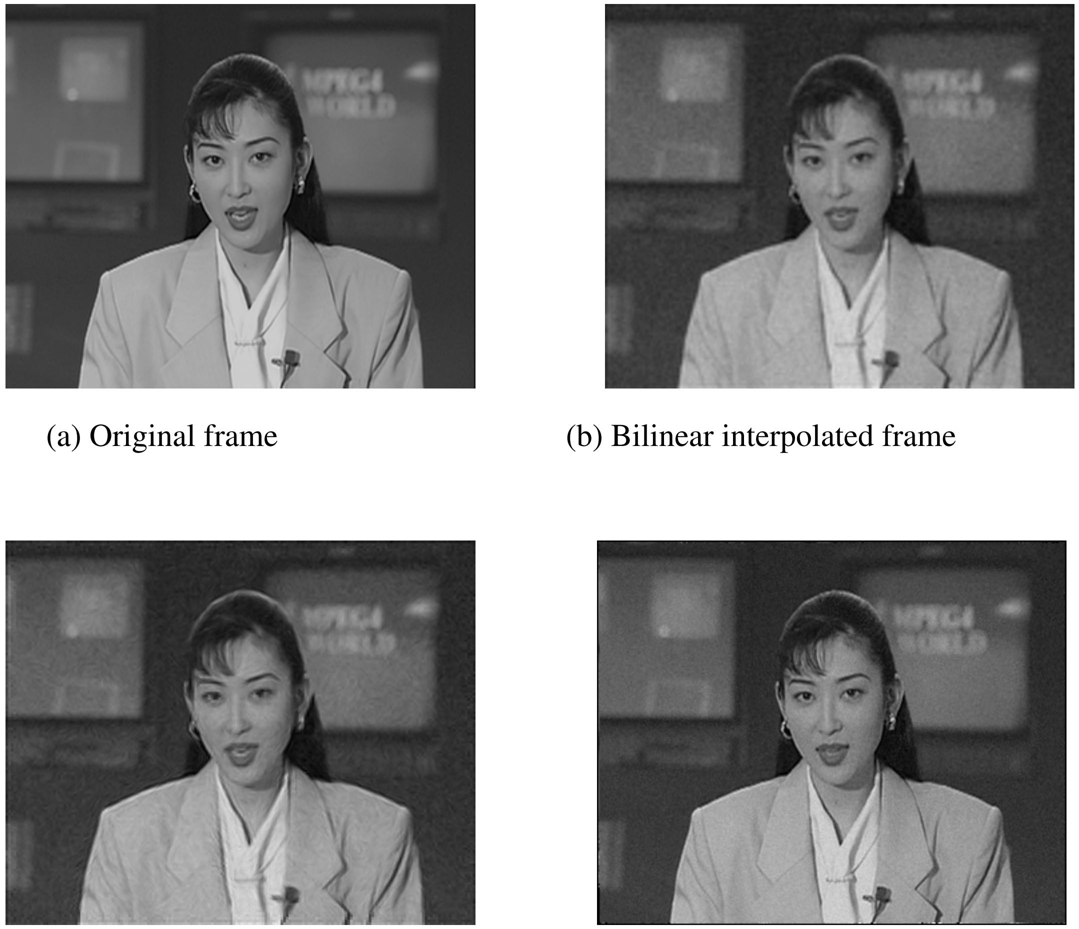
(c) Interpolated frame using NEDI

(d) Interpolated frame using full band warping

Fig. 3.13. Comparison of the subjective quality for the noisy Akiyo sequence using full band warping and its competing methods. The added noise is additive Gaussian noise with mean of 0 and standard deviation of 2. The frame shown is the 13th frame in the sequence.

3.5 High Frequency Warping

Although the full band warping method achieves significant visual improvement over both the bilinear interpolation method and NEDI [6] , and the results have better sharpness and clarity compared to the original CIF frame, for some sequences the warped frames can display geometirc distortion, as illustrated in Figure 3.16. This occurs typically when



(a) Original frame

(b) Bilinear interpolated frame

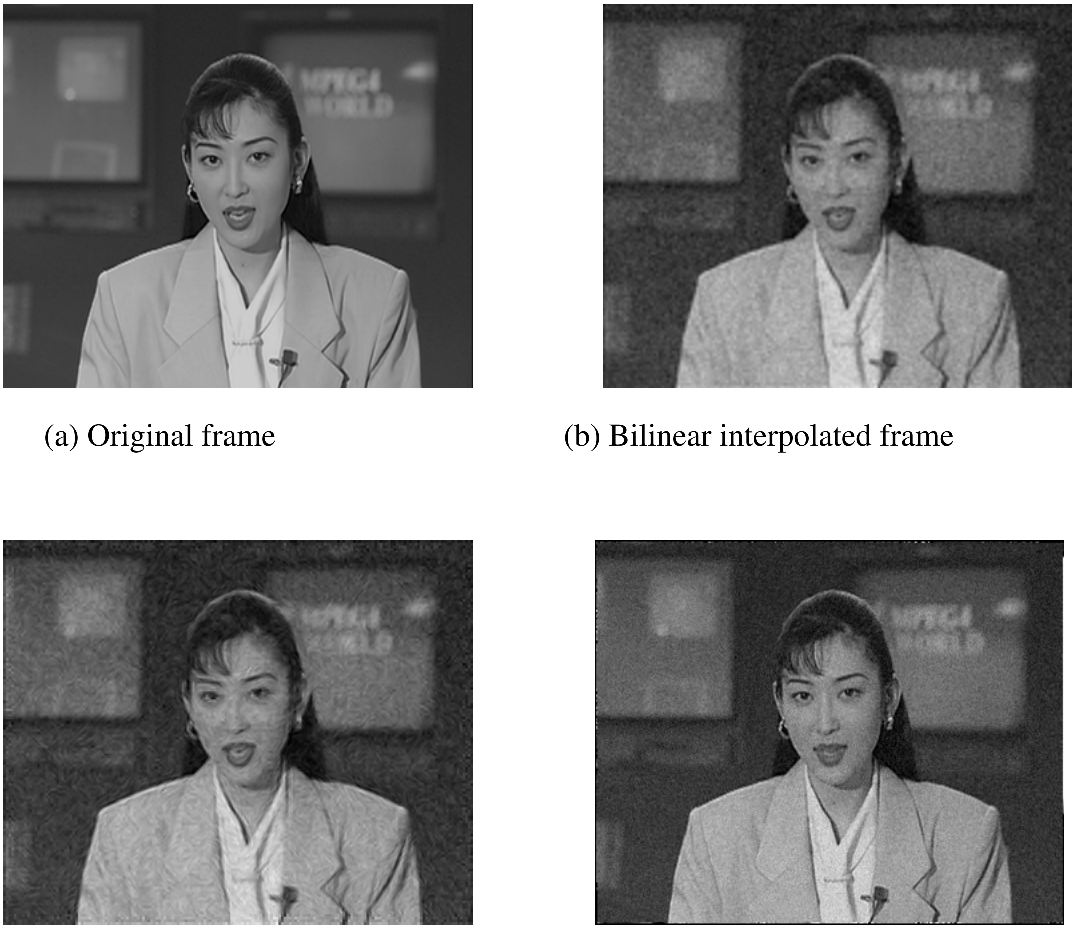
(c) Interpolated frame using NEDI

(d) Interpolated frame using full band warping

Fig. 3.14. Comparison of the subjective quality for the noisy Akiyo sequence using full band warping and its competing methods. The added noise is additive Gaussian noise with mean of 0 and standard deviation of 5. The frame shown is the 13th frame in the sequence.

large local motion is presented and the motion vector accuracy is poor. As we can see, the nature of the distortion is not loss of sharpness but rather a twisting of contours and edge boundaries.

In the context of our method, the fundamental issue is fusing the high definition spatial information into the low resolution frames. This perspective motivated the high frequency warping approach which is presented next.



(a) Original frame

(b) Bilinear interpolated frame

(c) Interpolated frame using NEDI

(d) Interpolated frame using full band warping

Fig. 3.15. Comparison of the subjective quality for the noisy Akiyo sequence using full band warping and its competing methods. The added noise is additive Gaussian noise with mean of 0 and standard deviation of 10. The frame shown is the 13th frame in the sequence.

By downsampling the high quality $2N \times 2M$ reference frame, and then interpolating it back to $2N \times 2M$, we effectively equalized both images with respect to their frequency resolution. By warping the low resolution reference frame into the interpolated subsequent video frames, we obtain motion vectors that capture the movement of the objects in the video sequence. This helps to assure that local motion vector accuracy is sufficient to avoid visible geometric distortion.

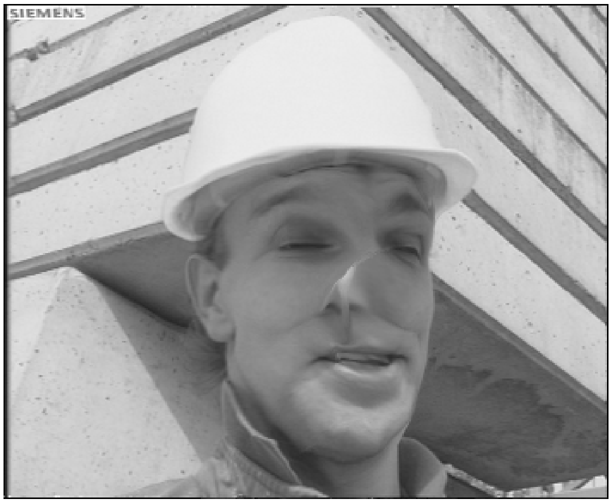


Fig. 3.16. Frame taken from the Foreman sequence that illustrates geometric distortion that can occur in the full band warping method.

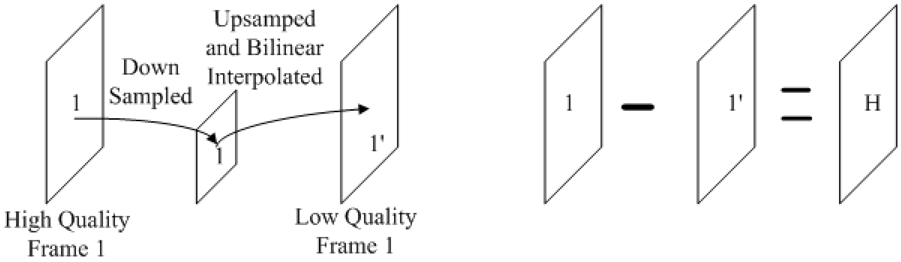


Fig. 3.17. First step of high frequency band warping

The high frequency warping method consists of four parts. The first part is to obtain the high frequency components in the reference frame illustrated by Figure 3.17. The second step is to obtain the motion vectors by warping the reference frame without the high frequency components into the interpolated low quality $2N \times 2M$ size frames. The third step is to apply those motion vectors to the high frequency components obtained from the first step. In the fourth step, we add the lowpass high resolution frames and warped high frequency component to get the fullband frames. Steps 2, 3 and 4 are depicted in Figure 3.18. In step 1, we first downsample the high quality reference frame called A, and then bilinearly interpolate it back to $2N \times 2M$. We denote it as Y_1. By subtracting the

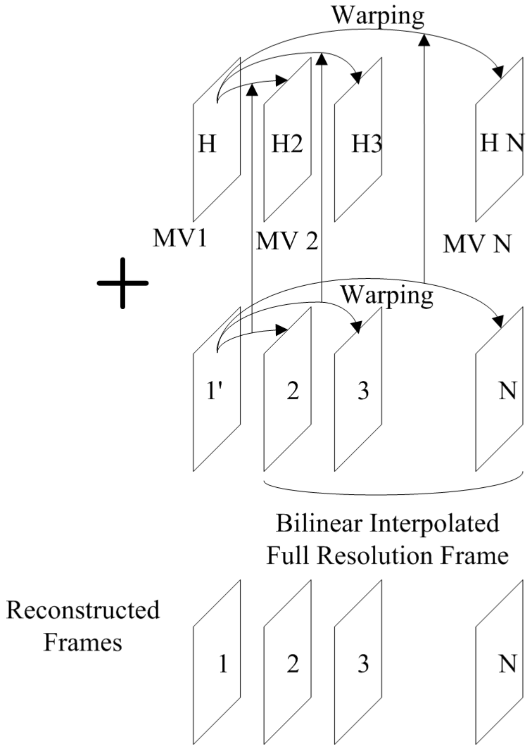
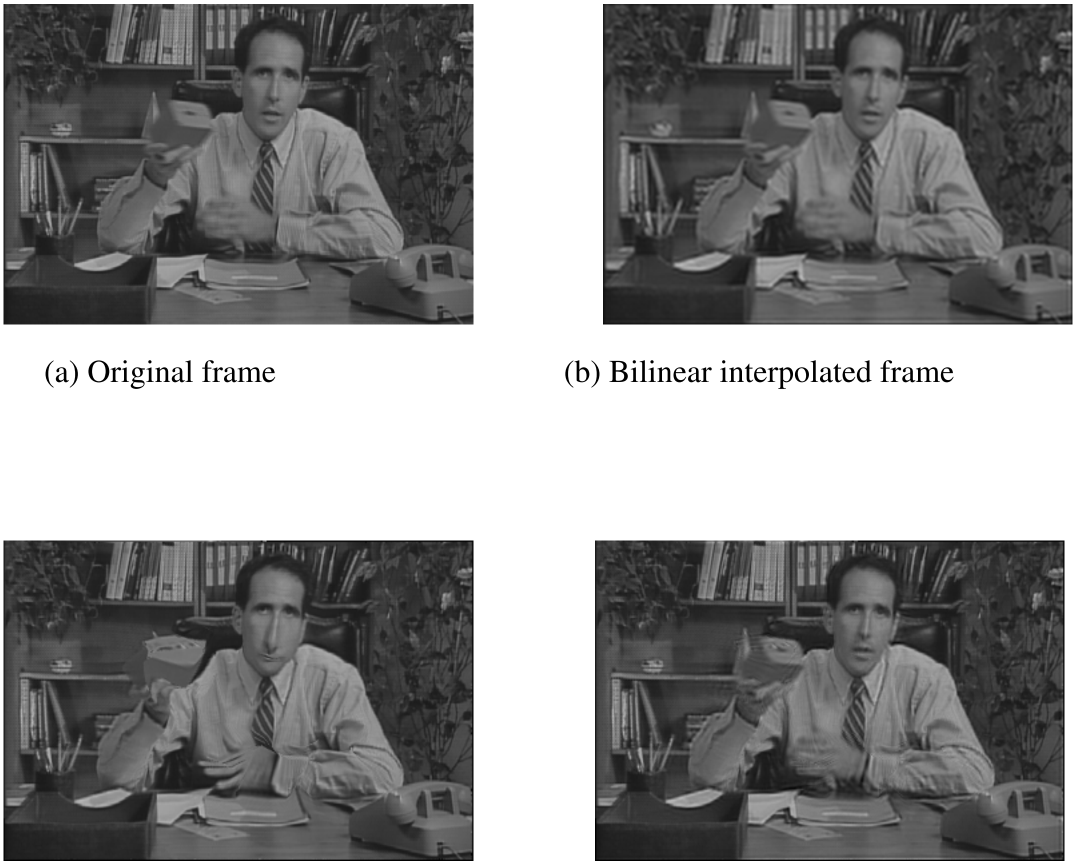


Fig. 3.18. Step 2, 3, 4 of high frequency band warping

lowpass $2N \times 2M$ size frame, called Y_1, from the high quality reference frame A, we get the high frequency components H. In step 2, we warp Y_1 into subsequent lowpass bilinearly interpolated high resolution frames and get the motion vectors, called MV(i), $i = 2, ..., N$. In step 3, we apply those motion vectors MV(i) to the high frequency component H and then get $H(i), i = 2, ..., N$, for the subsequent frames. Finally, in step 4, we add the lowpass high resolution frames $X'(i)$, resulting from direct interpolation, to the warped high frequency component $H(i)$ to get the full band frames $F(i)$. In other words, $X'(i) + H(i) = F(i)$.

3.5.1 Experimental Results

The high frequency warping algorithm was tested on the talking head lecture video sequences: Akiyo, Foreman, Carphone, News, and Salesman. The way we tested the se-



(a) Original frame

(b) Bilinear interpolated frame

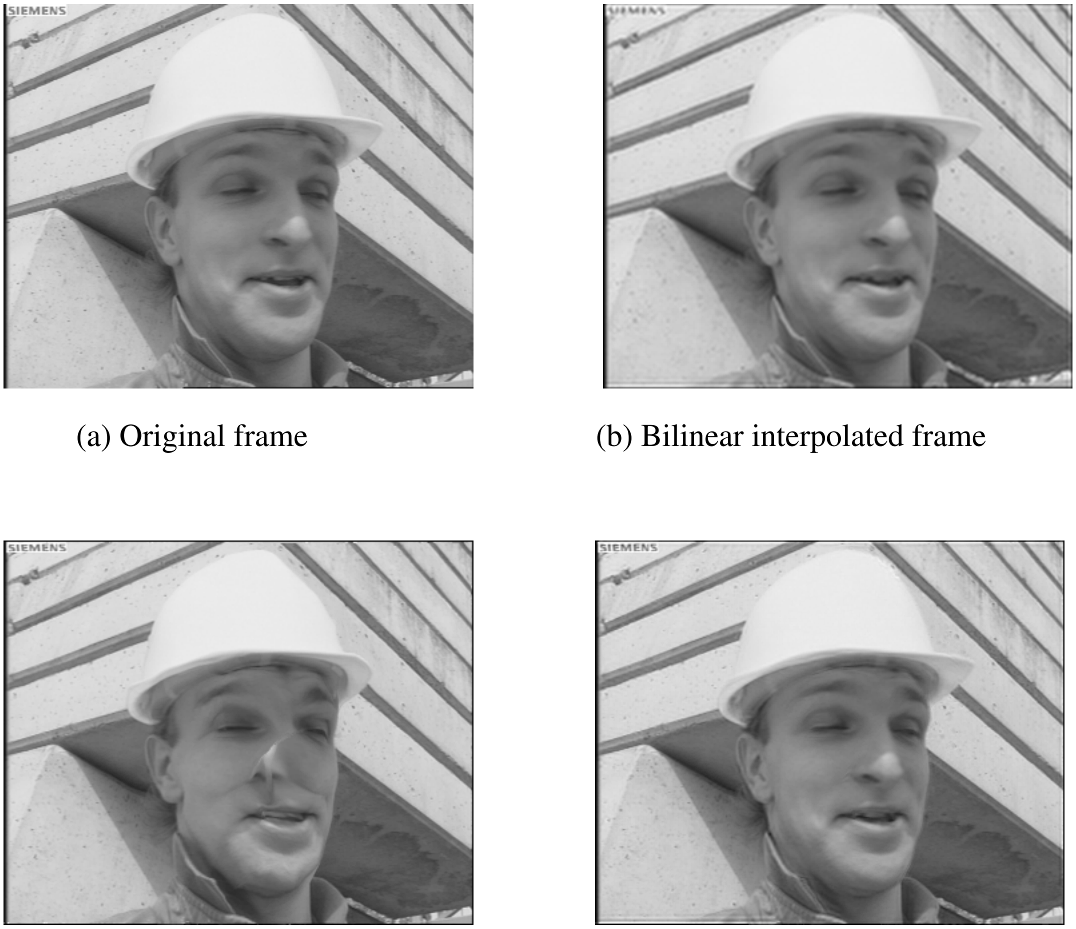
(c) Frame interpolated using full band warping

(d) Result using high frequency warping

Fig. 3.19. Comparison of the subjective quality for the Salesman sequence using full band warping and high frequency warping. The frame shown is the 10th frame in the sequence.

quences is the same as what we did for the full band warping, that is, keep the first CIF size frame and downsample the subsequent frames to get the low resolution video frames.

In Figures 3.19 - 3.23, we show the subjective experimental results for those sequences. We show two types of results: first the robustness of the high frequency warping algorithm over full band warping, and second the performance quality of full band warping and high frequency warping when the full band warping method does not break down.



(a) Original frame

(b) Bilinear interpolated frame

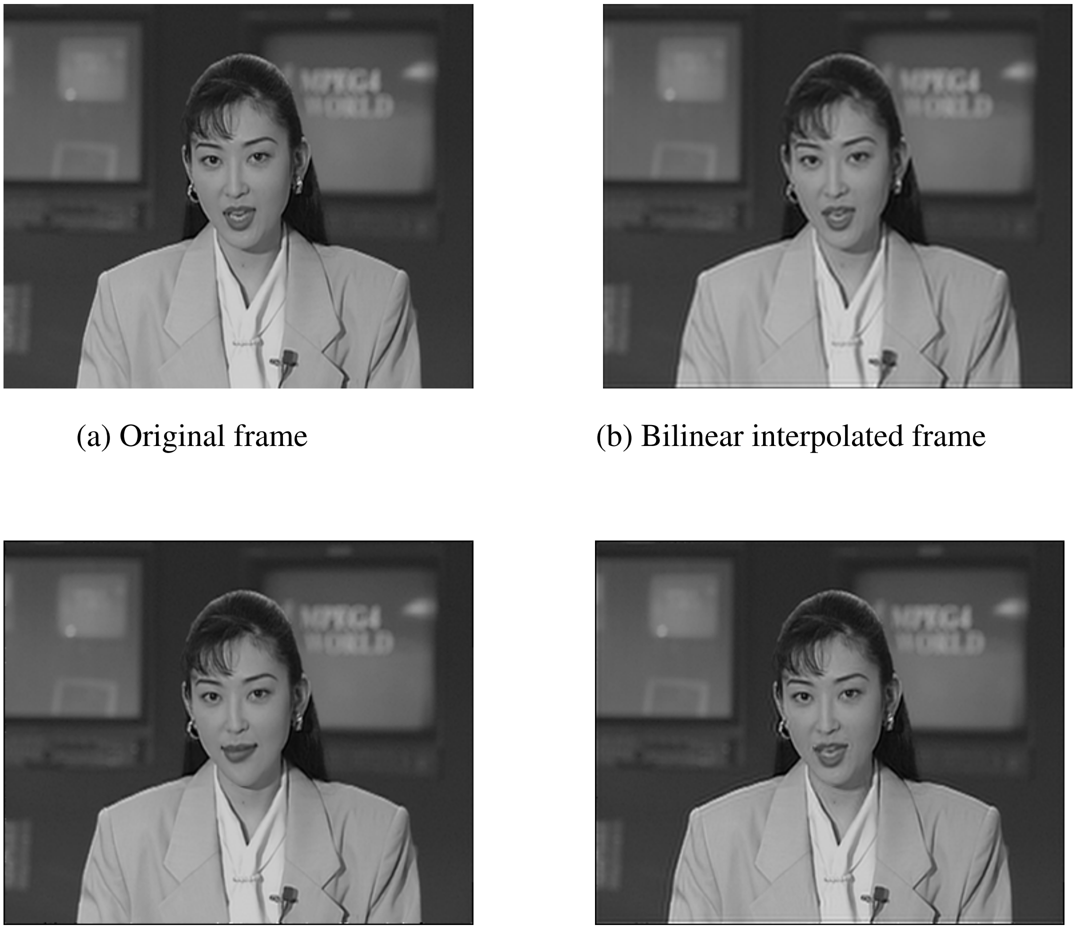
(c) Frame interpolated using full band warping

(d) Result using high frequency warping

Fig. 3.20. Comparison of the subjective quality for the Foreman sequence using full band warping and high frequency warping. The frame shown is the 3rd frame in the sequence.

For each sequence, we show first the original $2N \times 2M$ size frame and then the bilinearly interpolated frame followed by the full band warped frame and high frequency warped frame.

Figures 3.19 - 3.23 show the subjective results, where the full band warping method breaks down but the high frequency warping methods produces reasonable results and achieves better visual results than bilinear interpolation.



(a) Original frame

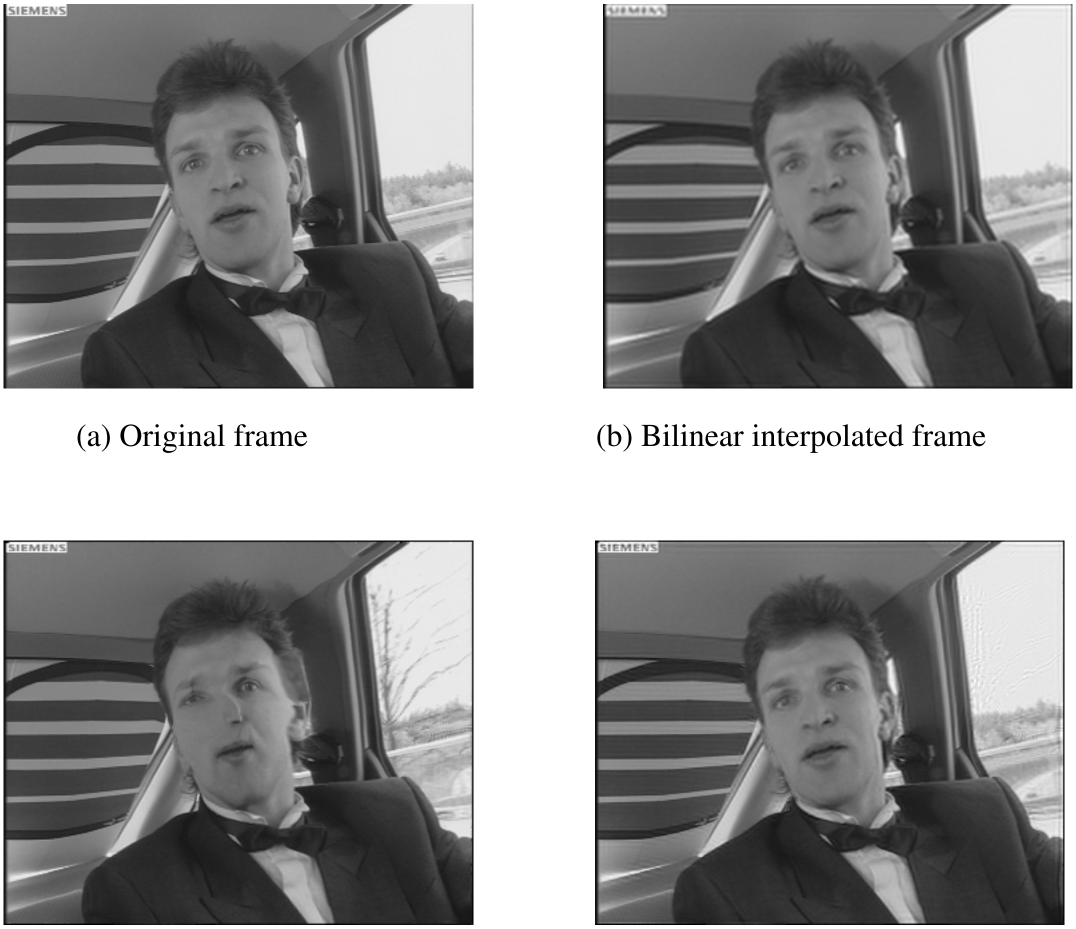
(b) Bilinear interpolated frame

(c) Frame interpolated using full band warping

(d) Result using high frequency warping

Fig. 3.21. Comparison of the subjective quality of frames from Akiyo sequence using full band warping and high frequency warping. The frame shown is the 13th frame in the sequence. Notice that the lips are closed in (a), a consequence of geometric distortion.

Although the high frequency warping method tends to be robust and free of geometric distortion, some of the conventional distortions can be seen. As we notice when the full band warping method is working acceptably, its overall visual quality is better than the high frequency warping method . This point is illustrated in Figure 3.24-3.25. In particular,notice there are ringing artifacts in the high frequency warped frames.



(a) Original frame

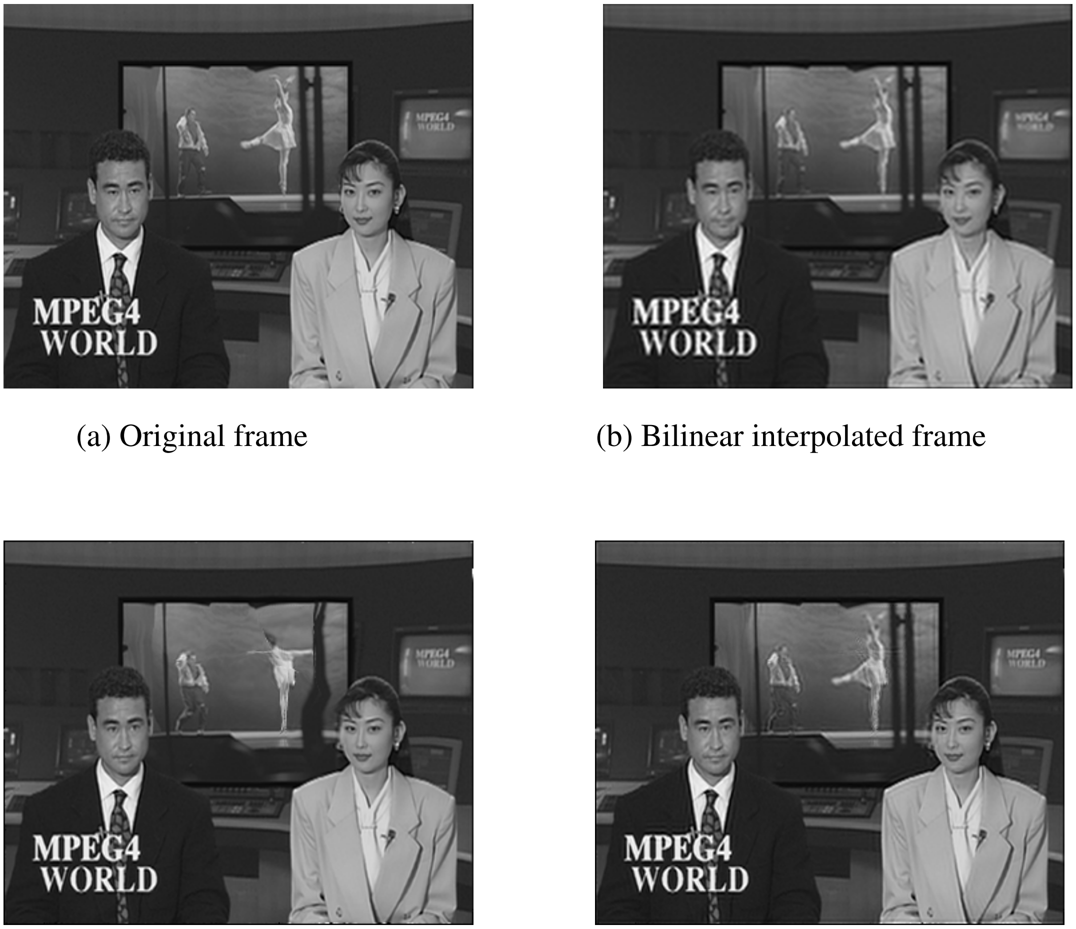
(b) Bilinear interpolated frame

(c) Frame interpolated using full band warping

(d) Result using high frequency warping

Fig. 3.22. Comparison of the subjective quality for the Carphone sequence using full band warping and high frequency warping. The frame shown is the 28th frame in the sequence. The geometric distortion is obvious in (c).

The major problem in the high frequency warping method is that because it uses the bilinearly interpolated frame as a basis to be robust and adds the warped high frequency component to make the output sharp, it inherits the ringing artifacts from the bilinear interpolation method.



(a) Original frame

(b) Bilinear interpolated frame

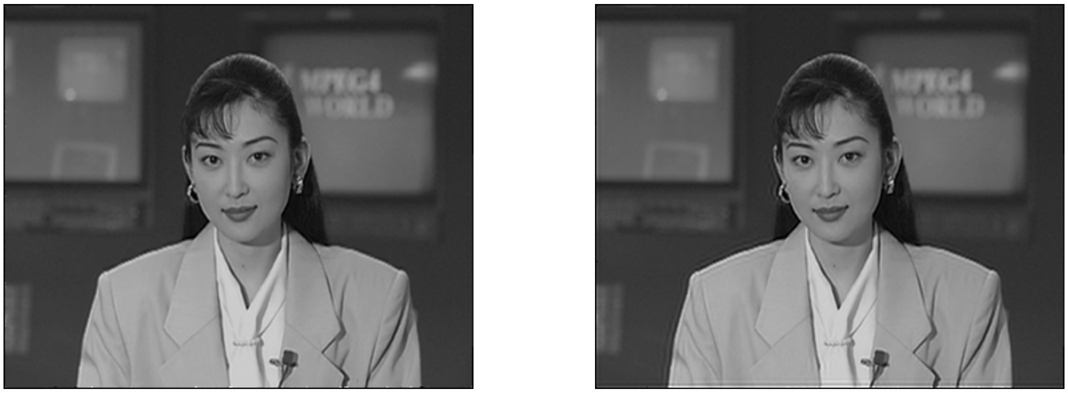
(c) Frame interpolated using full band warping

(d) Result using high frequency warping

Fig. 3.23. Comparison of the subjective quality for the News sequence using full band warping and high frequency warping. The frame shown is the 10th frame in the sequence. Notice the geometric distortion in the dancers in (c).

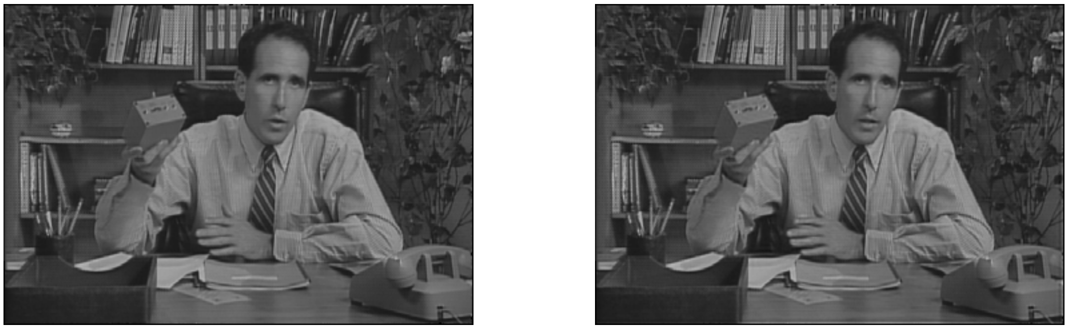
3.6 Summary

As we can see from the results presented, both the full band warping and high frequency warping methods can suffer quality degradation. When there are corresponding pixels in the reference (i.e. low motion), the subjective result for the resulting interpolated frames are satisfactory and frames appear sharp. But when the high local motion is present, there



(a) Warped frame using full-band warping (b) Warped frame using high frequency warping

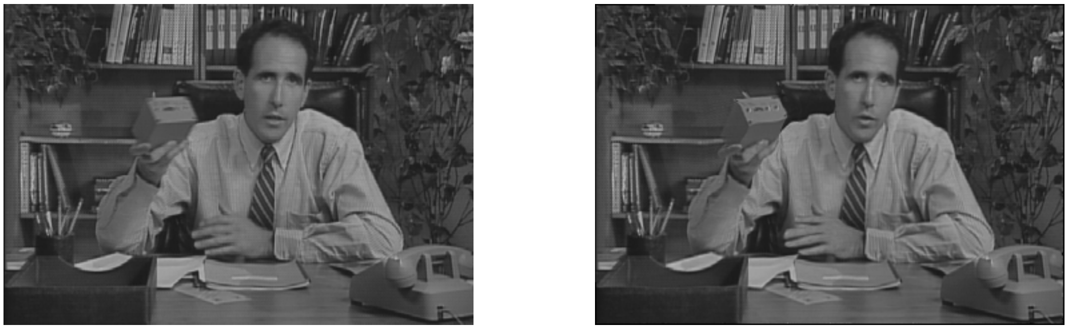
Fig. 3.24. Algorithm comparison for the 3rd frame of the Akiyo sequence



(a) Warped frame using full-band warping (b) Warped frame using high frequency warping

Fig. 3.25. Algorithm comparison for the 3rd frame of the Salesman sequence

is more chance that there are no corresponding pixels existing in the reference frame and the warped frame gets distorted. Furthermore, the warped frames resulting from the full band warping are not necessarily an accurate approximation of the original frame. This can be seen in Figure 3.21(c). Notice that the quality and sharpness of the image are high, and the image looks very good. But the lips of the speaker are closed, where in fact they should



(a) Original frame (b) Warped frame using full-band warping

Fig. 3.26. Algorithm comparison for the 7th frame of the Salesman sequence to show the difference between the original frame and the warped frame

be open. Another example is shown in Figure 3.26. Here the quality is quite acceptable, but the full band warped image is slightly different.

Practically speaking, as long as the visual quality is sharp and appear natural to the eye, it satisfies our objective.

The challenge for this approach is preserving accuracy when pixel displacements are large as in the case of high motion. Better performance is anticipated by allowing greater displacement flexibility in the warping algorithm. This potentially can be achieved by applying more robust motion models that better account for the mapping between the reference and the current frame.

In Chapter 4 we will introduce a more robust way to do the warping. The main idea there is to try to avoid those erroneous motion vectors that result from occlusion or other situation where no corresponding pixels exist in the reference frame.

3.7 Acknowledgments

Part of the text in this chapter is reprinted from the paper “High Quality Spatial Interpolation of Video Frames Using an Adaptive Warping Method”, which has been published in IEEE 12th Digital Signal Processing Workshop and 4th IEEE Signal Processing Education Workshop 2006 (DSP/SPE 2006) [52]. The dissertation author was the primary researcher and author, and the listed co-author in the publication supervised and directed the research and also contributed to this chapter.

4. THE COMPOSITE ALGORITHM

The full-band warping method we proposed in Chapter 3 produces satisfactory results when the motion in the sequence is low to moderate. But when the motion is high, the full band warping method breaks down, resulting in visible geometric distortion. Even though the high frequency warping method is efficient in mitigating geometric distortion, it does so at the expense of high spatial definition. Moreover, there can be ringing artifacts in the warped frames using the high frequency warping method.

In this chapter, we extend the method proposed in Chapter 3 to be bidirectional and using a new hierarchical motion structure to provide intelligent adaptivity in the processing. The experimental results are presented at the end, which validate the expected improvement in performance.

4.1 Forward and Backward Warping

In Chapter 3, the advantages and disadvantages of full-band warping were highlighted. The primary source of the performance shortcomings is inaccuracy in the estimated motion field. Such inaccuracies often arise when motion with a region is high, when occlusions occur, and when background is uncovered.

To address this issue, we exploit the presence of the updated reference frames. More specifically, forward warping is applied between the first reference frame and the current frame. Then backward warping is applied between the current frame and the second reference frame. This provides two high definition warped versions of the current frame. Because uncovered background in the forward motion estimation process is typically available in the backward motion estimation process and vice versa, using both high definition images together can dramatically reduce warping errors.

We illustrate the process in Figures 4.1 and 4.2. First (step 1) bilinear interpolation is performed on the subsequent low resolution $N \times M$ frames so that all the frames are of the same size. Next (step 2) forward warping and backward warping are performed. After we obtain two warped frames, we need a mechanism to produce pixels from those available frames. In the next subsection we describe a method for choosing the pixels from available data.

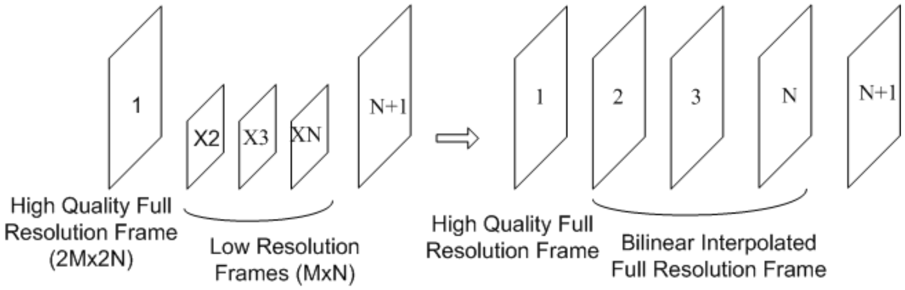

1
X2
X3
XN
N+1
High Quality Full Resolution Frame (2Mx2N)
Low Resolution Frames (MxN)
1
2
3
N
N+1
High Quality Full Resolution Frame
Bilinear Interpolated Full Resolution Frame

Fig. 4.1. Forward and backward warping (step 1)

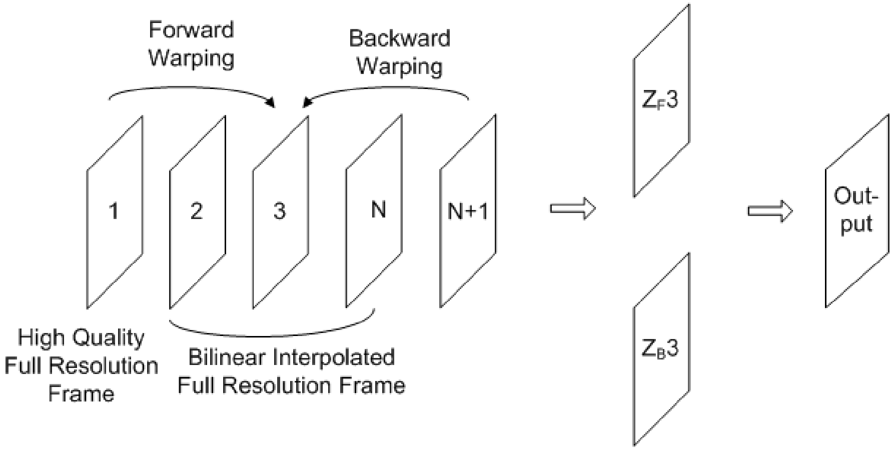

Forward Warping
Backward Warping
1
2
3
N
N+1
High Quality Full Resolution Frame
Bilinear Interpolated Full Resolution Frame
ZF3
ZB3
Out-put

Fig. 4.2. Forward and backward warping (step 2)

4.2 Hierarchical Motion Structure

In some cases, both forward and backward warped images may contain pixels in the same region that are corrupted. Thus, we consider a third choice, that of using an interpolated low resolution image.

Now with three enlarged images, all of the same frame, the task is to choose on a pixel-by-pixel basis the best from the three enlarged candidate frames. How to do this is an issue to address, considering that there is an inherent ambiguity associated with using a mean square error (MSE) approach. To elaborate on this point, consider that we compute the squared error between the warped high resolution frame and the interpolated frame over each block. One might expect that if the mean square error associated with that region were high, then geometric distortion has occurred. However, improvement in sharpness derived from the warping (when done correctly) will show up as error energy in the MSE. Consequently it may often be difficult to tell from the MSE if the source of the error is from geometric distortion or from low-frequency-high-frequency spatial differences.

To resolve this ambiguity, we perform the MSE calculations in the downsampled domain. In this domain, we don't have differences in sharpness contributing to the MSE. Thus, the composite algorithm is first to downsample the warped frame Z, which is $2N \times 2M$, so that it has the same resolution as the low resolution frames ($N \times M$).

In the low resolution domain, we compare the pixel values from the downsampled warped frame Z_d and the one from the low resolution video frame X. If the pixel value difference between Z_d and X is large, we assume that the corresponding 4 pixels in the original warped frame are not correct.

This idea is applied to both forward- and backward-warped images, given that every Nth frame will have a high definition reference frame. For each frame between any two reference frames, we first bilinearly interpolate the low resolution frame to high resolution and denote the result X_I. Then we use the preceding reference frame and warp it into X_I (forward warping) to obtain Z_F. Similarly, we use the succeeding reference frame and warp it to X_I (backward warping), resulting in Z_B. Generally speaking, when the current frame is temporally close to the preceding reference frame, the forward warped frame is most accurate. Likewise, when the current frame is close to the succeeding reference frame, the backward warped frame is typically better. After performing the block MSE calculations, we select Z_B or Z_F depending on which has the lower MSE. In the event that the block MSEs for both Z_B and Z_F are above threshold, we use the block derived from bilinearly interpolating X. While it is true that this block will lack the desired high frequency detail, the infrequency of our choosing the interpolated case combined with the blocks being relatively small leads to an overall enlarged image that appears sharp and void of geometric distortion. Figures 4.3 and 4.4 describes graphically the process detailed above. Figure 4.3 describes the bidirectional warping process while Figure 4.4 illustrates the pixel selection process as explained above.

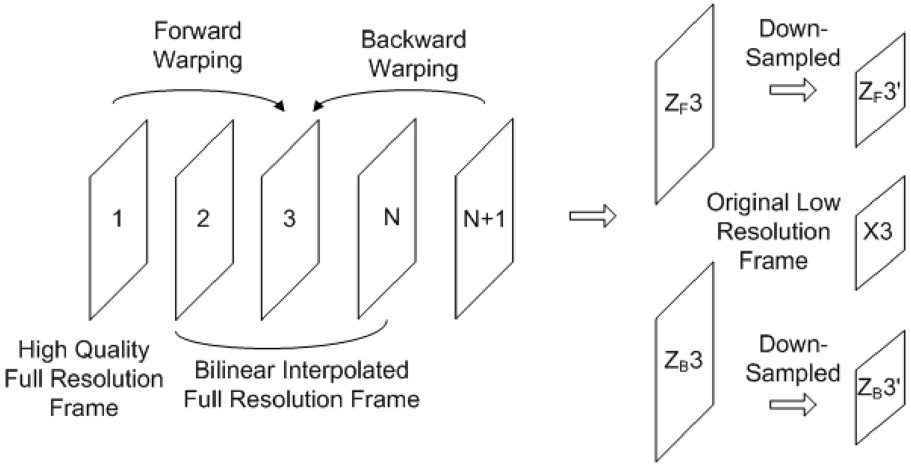
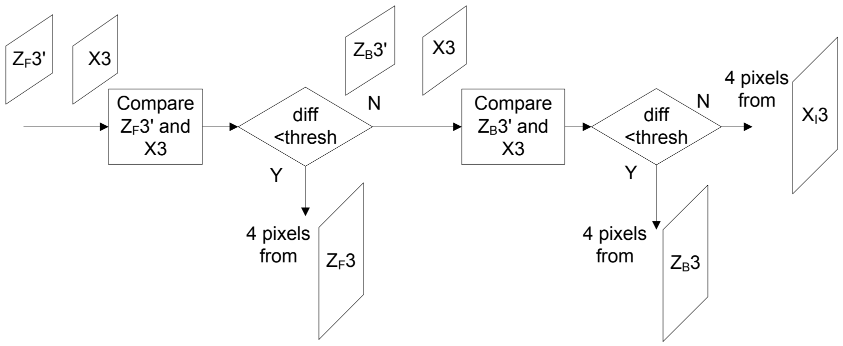

Forward Warping
Backward Warping
1
2
3
N
N+1
High Quality Full Resolution Frame
Bilinear Interpolated Full Resolution Frame
ZF3
Down-Sampled
ZF3'
Original Low Resolution Frame
X3
ZB3
Down-Sampled
ZB3'

Fig. 4.3. Forward and backward warping together with hierarchical motion structure

4.3 Experimental Results

4.3.1 The Composite Algorithm using Bilinear Spatial Interpolation

In this experiment, we applied the composite algorithm on three types of video sequences: “talking head” video sequences, which are characterized as having low motion;



Z_F3: forward warped Z_F3': downsampled forward warped
Z_B3: backward warped Z_B3': downsampled backward warped
X3: input low resolution X_I3: spatially interpolated

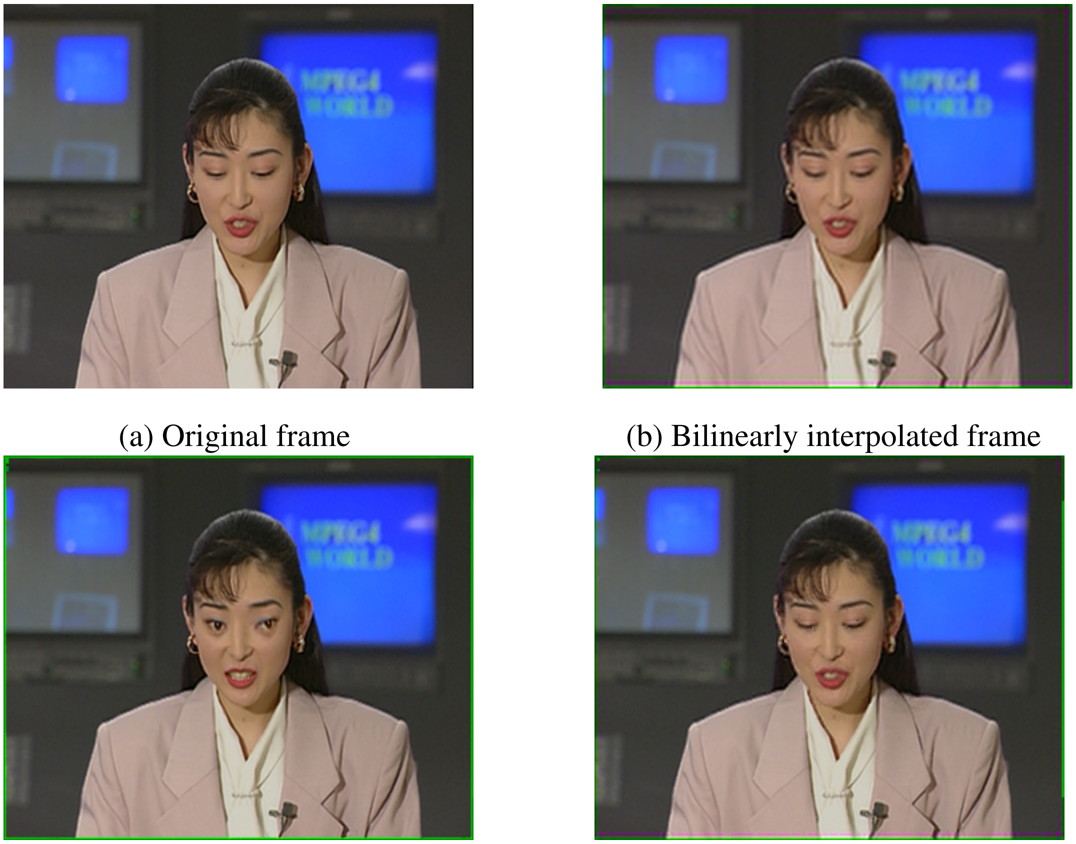
Fig. 4.4. Selection mechanism

sequences with high detail, such as Flower Garden and Tempete; and sequences with high motion, such as Football, Stefan and Table Tennis.

For each test sequence, every 10th frame was designated as a high definition reference frame. For the frames in between we first lowpass filter them and downsample them to get the low resolution frame. The downsample factor in this case is 2 in each direction. The low resolution frames and reference frames are used as the input. The warping process and pixel selection process described in the preceding section are then applied. The resulting enlarged output sequences are then compared to the original frames which serve as ground truth and from which we compute the PSNR. For each test sequence, 50 frames are considered in the comparison.

Some caution should be exercised in accepting the PSNR blindly as a measure of quality. The nature of the approach we've taken considers high definition warping features to be perfectly acceptable as long as the warping is not visually objectionable. Geometric features in the warped frames could be displaced by a pixel or two from what appears in the original. This, because it is not perceptible, is acceptable even though the PSNR will be reduced. As a result, we examine both subjective and PSNRs in our comparisons.

In Figures 4.5, 4.6, 4.7 we provide visual results for three sequences which represent three types of video sequences. They are Akiyo which represents talking head sequences, Bus which represents sequences with high detail, and Stefan which represents sequences with high motion. For each sequence, we plot the original frame, the bilinearly interpolated frame, the warped frame using the full-band warping method, and the interpolated frame from the proposed algorithm.



(a) Original frame

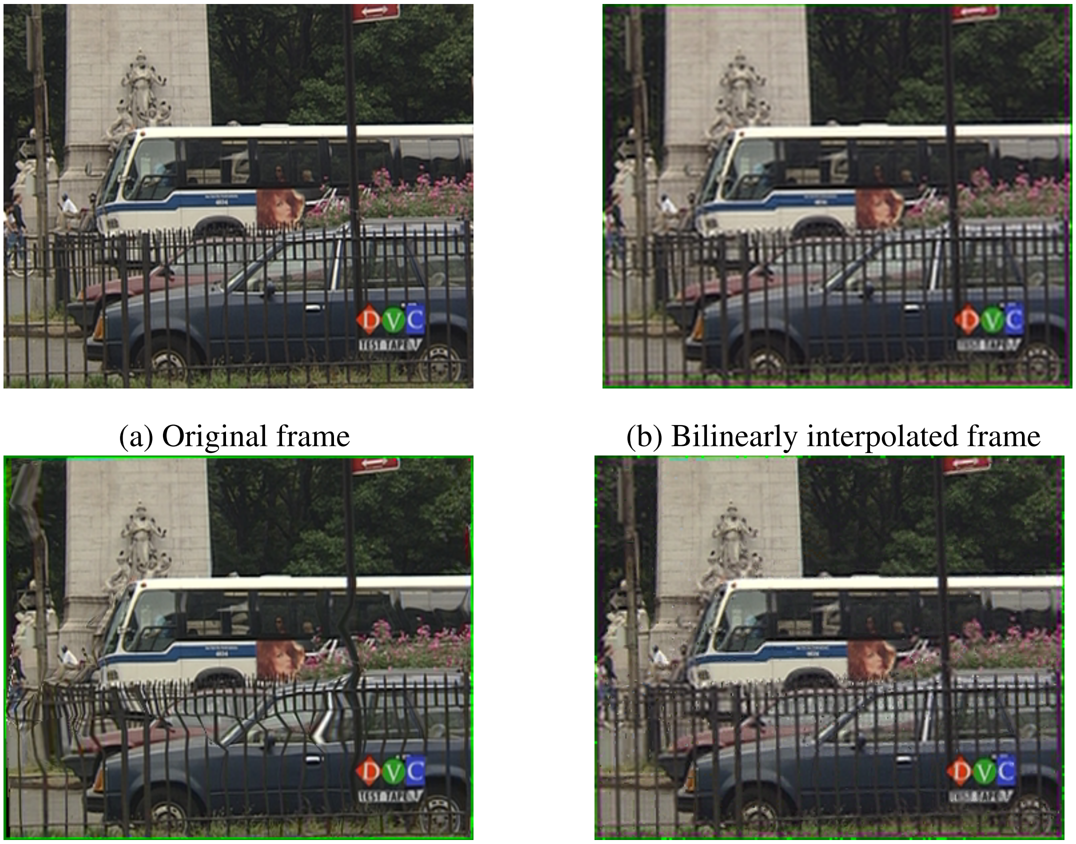
(b) Bilinearly interpolated frame

(c) Warped frame using full-band warping

(d) Warped frame using the composite algorithm

Fig. 4.5. Subjective comparison of the 39th frame of the Akiyo sequence. (a) Original frame. (b) Bilinearly interpolated frame. (c) Output of the warping method presented in Chapter 3. (d) Output of the method presented here.

As we can see from Figure 4.7(c), the warped frame overall looks very sharp but in some local areas, like around the left arm, there are geometric distortions. While the image and the distorted regions are all very sharp (compared to an interpolated frame in (b)),



(a) Original frame

(b) Bilinearly interpolated frame

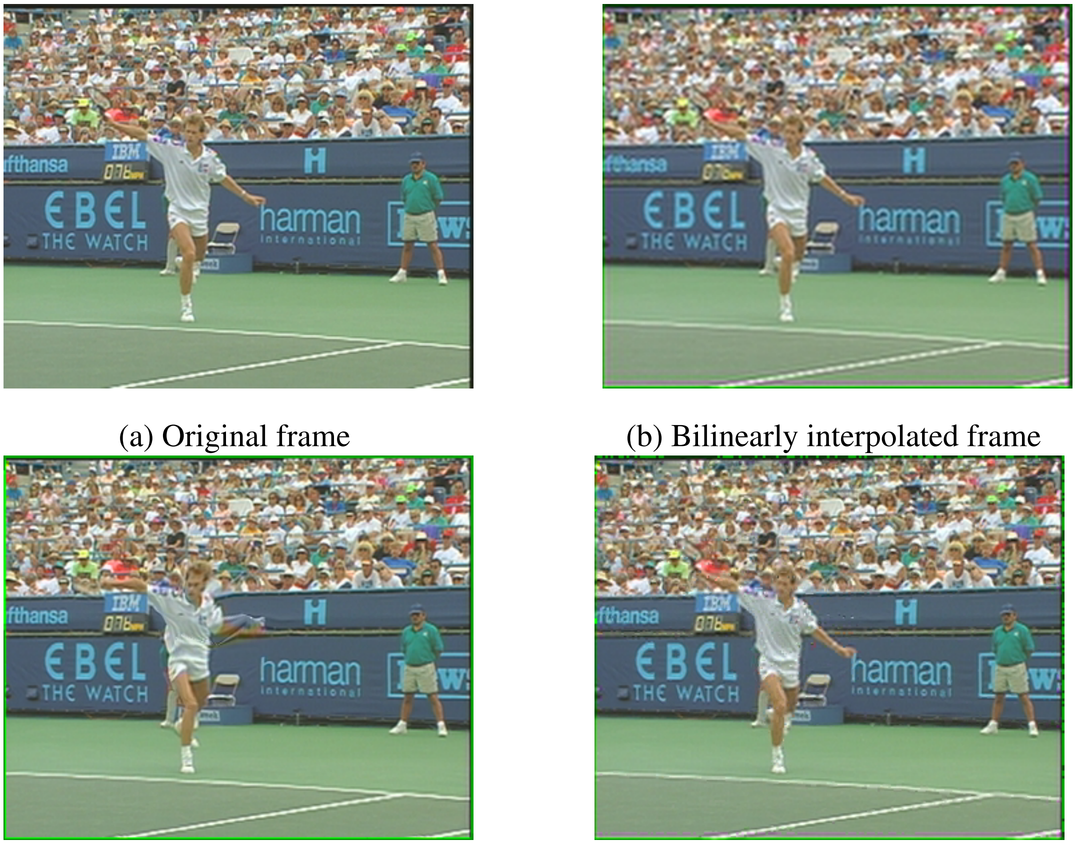
(c) Warped frame using full-band warping

(d) Warped frame using the composite algorithm

Fig. 4.6. Subjective comparison of the 32th frame of the Bus sequence. (a) Original frame. (b) Bilinearly interpolated frame. (c) Output of the warping method presented in Chapter 3. (d) Output of the method presented here.

the nature of the distortion is still objectionable. In contrast, the output of the proposed algorithm (Figure 4.7(d)) preserves the object geometry and provides a sharp enlargement.

Choice of the threshold can have an impact on the performance and can allow for a tradeoff between geometric distortion and spatial dispersion (i.e. interpolation blur). The lower the threshold the more biased the algorithm is toward selecting blocks derived from the interpolated frame. And the higher the threshold, the more the bias is directed toward the warped frames. In cases of a high threshold, one can sometimes observe small geometric distortions as well as improved PSNR. For the proposed algorithm we have generally



(a) Original frame

(b) Bilinearly interpolated frame

(c) Warped frame using full-band warping

(d) Warped frame using the composite algorithm

Fig. 4.7. Subjective comparison of the 28th frame of the Stefan sequence. (a) Original frame. (b) Bilinearly interpolated frame. (c) Output of the warping method presented in Chapter 3. (d) Output of the method presented here.

observed that the PSNR agrees with our subjective assessments when comparative tests are performed.

For comparative purposes, we present in the tables numerical PSNR assessments for our algorithm along with several competing methods. Shown in the tables are the PSNR results for bilinear interpolation, bicubic interpolation, the edge-directional interpolation method of Xin [6] (a.k.a NEDI), and the algorithm proposed in this paper. Tables 4.1 to 4.3 list the resulting PSNR for three sets of video sequences.

As we can see, for all three sets of video sequences, the proposed method outperforms bilinear interpolation, bicubic interpolation, and NEDI. For the talking head video

Table 4.1
PSNR comparison for talking head video sequences using different methods

	Akiyo	Salesman	Foreman	Carphone	News
Bilinear	33.33	29.59	29.64	30.33	28.57
Bicubic	34.07	30.12	30.11	30.76	29.50
NEDI	33.51	29.23	28.60	29.87	28.26
New	34.24	33.27	31.11	32.88	33.63

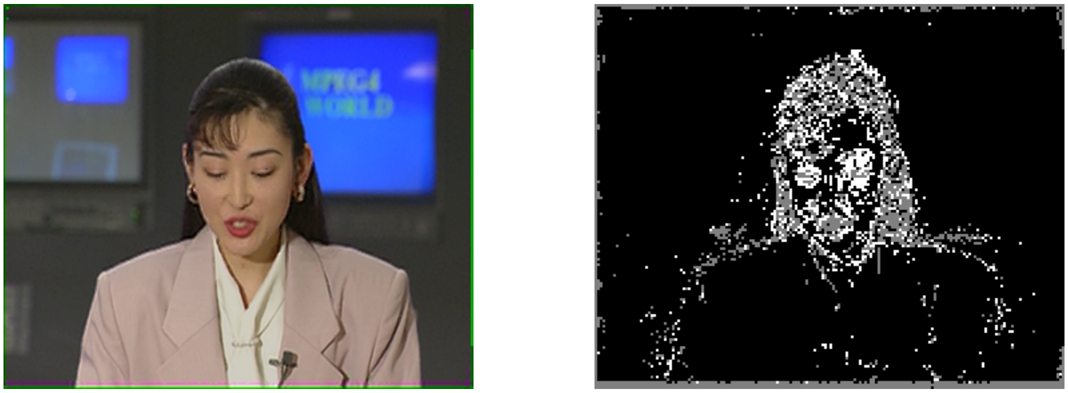
Table 4.2
PSNR comparison for video sequence with high detail.

	Salesman	Tempete	Flower	Bus
Bilinear	29.59	26.10	22.03	25.01
Bicubic	30.12	26.64	22.39	25.64
NEDI	29.23	25.57	21.41	24.48
New	33.27	29.40	26.77	26.04

Table 4.3
PSNR comparison for video sequence with moderate to high motion.

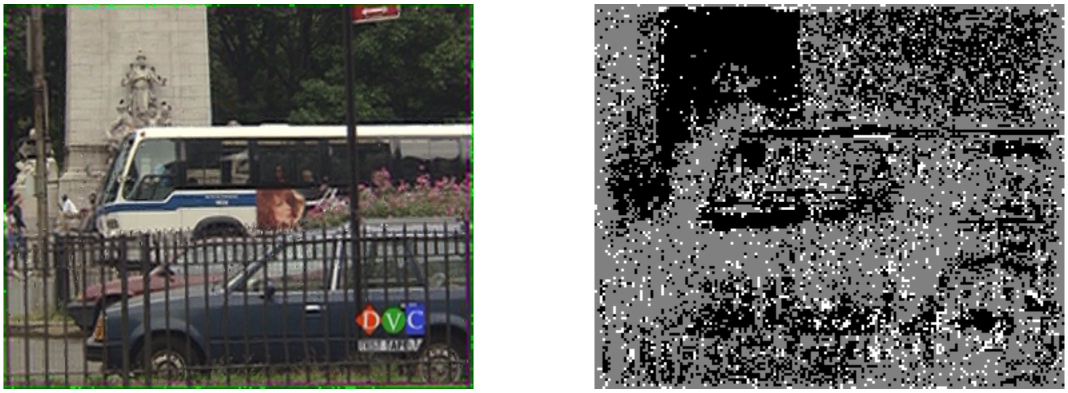
	Stefan	Table
Bilinear	25.57	29.06
Bicubic	26.33	29.76
NEDI	24.50	28.76
New	28.08	30.23

sequences, the composite algorithm outperforms the others by an average of 2 dB. For sequences with high detail such as Tempete and Flower, the composite algorithm achieves a 3-4 dB improvement on average. The reason that the composite algorithm performs particularly well for video sequences with high detail is that it inherently retains the high frequency information while such information is lost in methods that interpolate from a decimated frame.



(a) Output of the composite algorithm (b) Indicator of choices of motion vectors

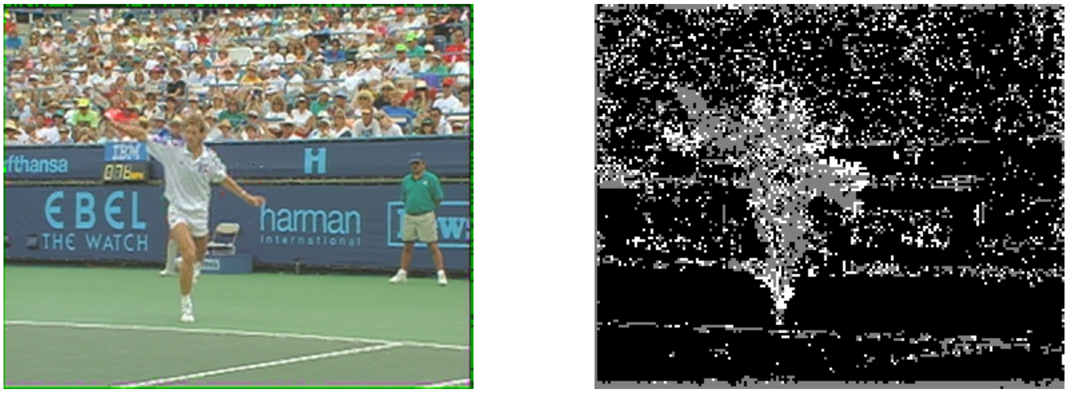
Fig. 4.8. Indicator of choice of motion vectors of the 39th frame of the Akiyo sequence.



(a) Output of the composite algorithm (b) Indicator of choice of motion vectors

Fig. 4.9. Indicator of choice of motion vectors of the 32th frame of the Bus sequence.

We also plot the choice of the motion vectors in Figures 4.8 to 4.10. The black pixels mean the pixel values were chosen from forward warping; white means they were chosen



(a) Output of the composite algorithm (b) Indicator of choice of motion vectors

Fig. 4.10. Indicator of choice of motion vectors of the 28th frame of the Stefan sequence.

from the backward warping; and gray means they were chosen from the bilinearly interpolated frame. As we can see from Figure 4.8(b), Figure 4.9(b), and Figure 4.10(b), the algorithm tends to choose the pixel from the bilinearly interpolated frame as the output near the area where there is high motion. In the area where there is less motion, it tends to choose the pixels from the forward warped frame. That is reasonable because the selection process favors forward warping in the absence of MSE error.

To test whether our proposed algorithm makes the right decision in choosing the pixel values, we count the one which is closest to the original pixel value as the ground truth among all three pixel candidates: the forward warped, backward warped, and bilinearly interpolated. We then compare the pixel values our algorithm chooses against the ground truth. If they are the same, we mark that pixel as the correct output. We calculate the correctness factor for all the sequences we tested in Tables 4.4 to 4.6. The correctness factor is calculated as

$$\text{correctness} = \frac{\#\text{of correct pixels}}{\text{total}\#\text{of pixels}}. \tag{4.1}$$

We noticed that for the talking head sequences, the correctness is much higher than the other two types of video sequences. It is understandable because for this type of sequence, most motion vectors obtained from the forward warping are correct and it is easier to make

Table 4.4
Correctness percentage of pixels chosen for talking head video sequences using the composite algorithm

Akiyo	Salesman	Foreman	Carphone	News
0.66	0.60	0.41	0.47	0.59

Table 4.5
Correctness percentage of pixels chosen for video sequences with high detail using the composite algorithm

Salesman	Tempete	Flower	Bus
0.60	0.33	0.45	0.18

Table 4.6
Correctness percentage of pixels chosen for video sequences with high motions using the composite algorithm

Stefan	Table
0.34	0.42

a decision. For the other types of video, the correcness factor is lower because there is more motion. But as we pointed out before, the visual improvement for the sequences with higher detail is more noticeable than the others. That is because there are more pixels to be evaluated in sequences with more detail. Thus the probability of being wrong is increased. In addition, we treat all the pixel values with the same importance. Usually in one image, the edges that have high definition have greater impact on the subjective quality than other pixels. So even if mistakes are made, the visual improvement is still higher than with the other type of sequences.

4.3.2 Influence of Spatial Interpolation on the Overall Performance

We considered the incorporation of other more sophisticated interpolation algorithms in place of simple bilinear interpolation in Stage 1 in Figure 3.1. In theory, using better component interpolations should translate to higher quality enlargement.

Here we consider three spatial interpolation methods along with one superresolution method. They are bilinear interpolation, bicubic interpolation, the new edge directional interpolation (NEDI) [6] and a frequency domain superresolution method (VA) [53]. The "new edge-directed interpolation" method proposed by Xin and Orchard [6] (NEDI) is already described in Section 3 of Chapter 3. The superresolution method proposed by Vandewalle et al. [53] is a frequency domain method built on the properties that a linear shift in space is equivalent to a shift in phase, and a rotation in space is related to an amplitude change in the polar coordinates in the frequency domain. High frequency components that may have incurred aliasing are discarded to improve the robustness. This method uses several low resolution frames to generate one high resolution frame and thus is not a purely spatial interpolation method. We list it here because later in the experiment section, it will be combined with our warping algorithm and used as the output from the spatial interpolation stage and as the input to the warping stage.

When implementing the VA algorithm, to have a fair comparison, we keep the high frequency information in the superresolution method. That is, we keep every Lth frame as the high resolution reference frame ($2N$x$2M$), and use simple bicubic interpolation to enlarge the remaining low resolution frames (NxM) to full resolution ($2N$x$2M$). Then the VA algorithm is used to interpolate the L frames with the same size as a group to enlarge the size, which is now $4N$x$4M$. Finally they are downsampled to $2N$x$2M$. Thus the high frequency detail is preserved. A total of eight different PSNR results (bilinear, bicubic, NEDI, VA, bilinear+composite, bicubic+composite, NEDI+composite, VA+composite) can be obtained for each video sequence. We tested three sets of different video sequences which are all CIF size. The first set of video sequences are "talking head" video sequences. Those include the Akiyo, Silence, Mother and Daughter, and News sequences. The second set of video sequences have high texture and include the Salesman, Bus, Flower Garden, Tempete and Mobile sequences. The third set of sequences are Stefan, Table Tennis and Football which are characterized as containing moderate to high motion.

In Table 4.7 - 4.9, the PSNR results for three sequence sets computed for 20 frames from each sequence are listed. We use abreviations Bi-Comp, Cu-Comp, NEDI-Comp, VA-Comp to denote the embedded bilinear interpolation, bicubic interpolation, NEDI and VA methods within the composite algorithm. The PSNR reported is the average for the luminance component. Each fifth frame is chosen as the high resolution reference frame and subsequent frames are first lowpass filtered and then downsampled to obtain their low resolution frames. The original frames are used as the ground truth to calculate the PSNRs. Border effects that appeared in the top 5 rows and 5 right columns were cropped in all cases, since we have not yet implemented the symmetric extension method to address these few pixels.

Table 4.7
PSNR comparison for talking head video sequences using different spatial interpolated methods.

	Akiyo	Carphone	News	Silence	Mother
Bilinear	34.23	32.3	28.86	32.99	36.46
Bicubic	35.08	32.92	29.85	33.64	37.45
NEDI	34.98	32.64	28.7	32.76	36.58
VA	35.08	32.91	29.86	33.65	37.47
Bi-Comp	42.09	34.84	36.17	38.2	40.54
Cu-Comp	43.15	35.08	37.04	38.34	40.89
NEDI-Comp	41.24	34.14	35.23	36.98	39.53
VA-Comp	43.13	35.08	37.04	38.24	40.68

As we can see from Tables 4.7, 4.8, and 4.9, the spatial interpolation methods do influence the composite algorithm results. That is, if the spatial interpolated frame has higher quality, the warped frame also tends to perform better than its counterpart. In general,

Table 4.8
PSNR comparison for video sequences with high detail using different spatial interpolated methods.

	Sales	Bus	Flower	Tempete	Mobile
Bilinear	30.07	25.84	23.07	26.66	22.39
Bicubic	30.63	26.55	23.47	27.23	22.94
NEDI	30.05	25.35	22.71	26.34	22.11
VA	30.63	26.56	23.47	27.23	22.94
Bi-Comp	34.96	25.7	27.38	30.28	25.66
Cu-Comp	35.31	25.85	27.92	31.15	25.79
NEDI-Comp	34.21	25.25	27.13	30.15	25.79
VA-Comp	35.3	25.85	27.9	31.15	26.93

Table 4.9
PSNR comparison for video sequences with moderate to high motion using different spatial interpolated methods.

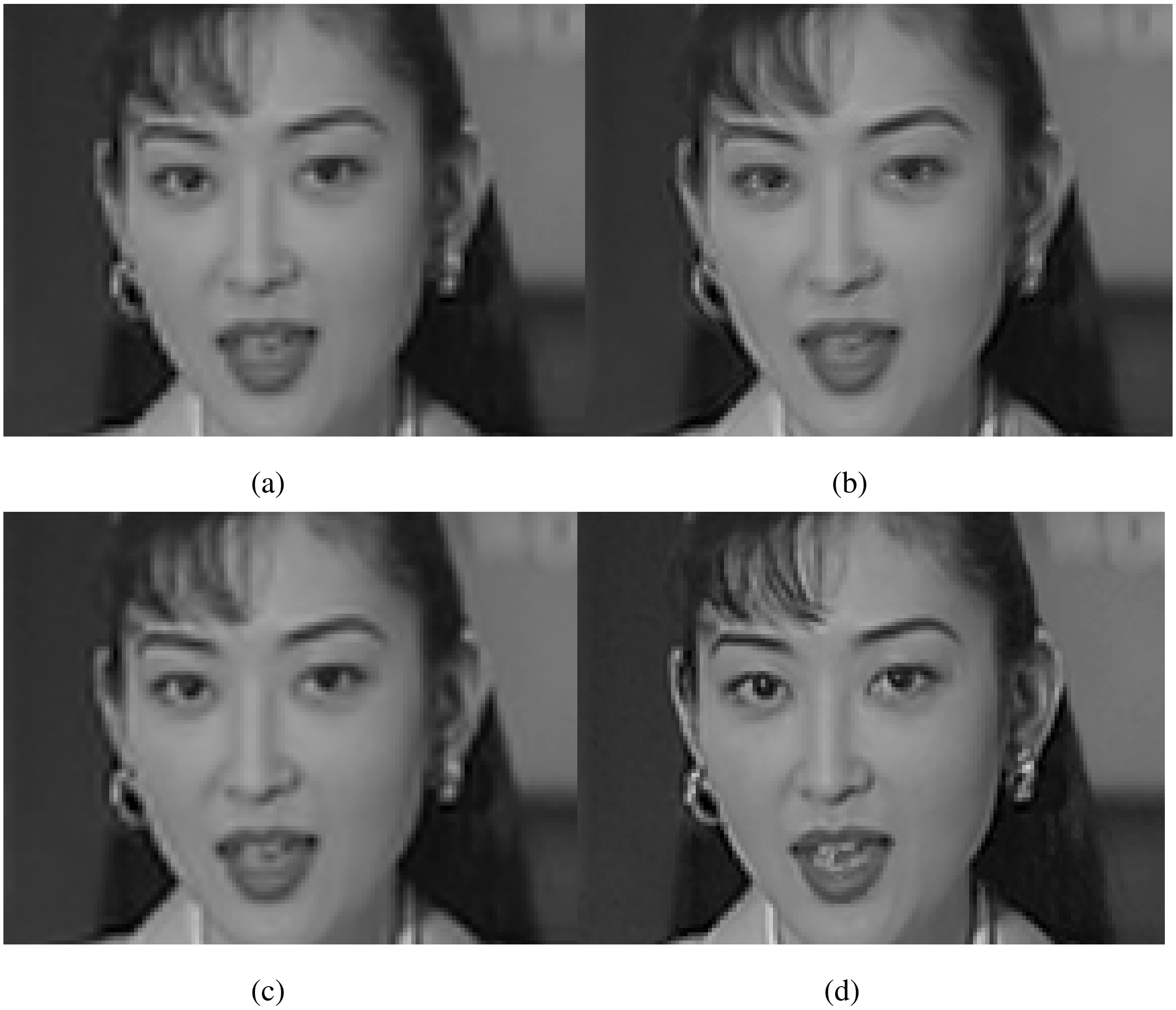
	Stefan	Table Tennis	Football
Bilinear	26.56	29.2	28.47
Bicubic	27.45	29.87	29.54
NEDI	25.91	29.25	28.14
VA	27.45	29.88	29.54
Bi-Comp	26.59	30.55	27.26
Cu-Comp	26.69	30.75	27.07
NEDI-Comp	26.12	30.15	26.48
VA-Comp	26.69	30.73	27.08

VA, Bicubic, NEDI and Bilinear interpolation are ranked in descending order in terms of PSNRs, among which VA and bicubic interpolation achieve very close performance. Thus VA-Comp and Cu-Comp also achieve similar results, both superior to Bi-Comp and NEDI-Comp in general. Because the VA superresolution method has a much higher computational complexity than the bicubic interpolation method, in term of the efficiency and performance, the use of bicubic interpolation in the composite algorithm is probably the best choice among the four.

For "talking head" video sequences, no matter which spatial interpolation algorithms is used, the improvement of the warping method over direct spatial frame interpolation is large in general, ranging from 3dB to 7dB. For Akiyo, News, and Mother & Daughter sequences, the difference between four spatial algorithms achieve an average of 0.8dB. For video sequences rich in texture, the average improvement is 4dB. The maximum differences among the spatial interpolation methods embedded in the composite algorithm are also noteworthy. For Mobile, Tempete and Flower Garden sequences, the differences are 1.27dB, 0.87dB, 0.52dB, respectively. For the third set of video sequences which has moderate to high motion, the differences are very small. This is partly because the differences in results among the spatial interpolation methods are small and when the motion is high, most pixels are extracted from the spatial interpolation reconstruction. In addition, the warping method itself doesn't produce very reliable motion vectors when the motion is high. Thus the pixel selection from the forward warping and backward warping is quite random. It is not uncommon for the composite algorithm to obtain lower PSNRs in high motion videos, as illustrated in Table 4.9. Even though subjective results generally are better, the composite algorithm is most appropriate for sequences with low to moderate motion.

4.4 Summary

In this chapter, the composite algorithm is introduced to improve the accuracy of the detected motion fields. When there are no pixels found in the preceding reference frame or when occlusions occur, the corresponding pixels can be found in the succeeding reference frame. The selection is done in the downsampled domain where the MSE represent the true difference more precisely. Later on, a concatenated model consisting of frame interpolation, warping and post-processing stages was introduced. We compared and in-



(a) (b)

(c) (d)

Fig. 4.11. Interpolation results from the different interpolation methods for the 17th frame of the Akiyo sequence. (a) Bilinear interpolation; (b) NEDI; (c) VA; (d) Bicubic+Composite.

tegrated several spatial interpolation methods into our warping algorithm as well as one spatial domain superresolution method. Experimental results show that the composite algorithm improves the objective and subjective quality significantly on sequences with low motion, regardless of which spatial interpolation methods are used. Better spatial interpolation methods do improve the overall output quality. However, the major gain comes from the adaptive warping itself. In terms of performance and complexity, the bicubic spatial interpolation followed by the warping algorithm produces the best tradeoff. For sequences with high texture, the objective improvements are not as large as those for low motion sequences. However, the subjective improvement is still high. For sequences with moderate to high motion, the gains diminish. This is because for high motion video sequences, the post-processing stage tends to choose pixels from the spatially interpolated frames.

4.5 Acknowledgments

Part of the text in this chapter is from a reprint of “A Robust Motion Estimation Method Using Warping for Video Frame Enlargement”, which has been published in the IEEE International Conference on Acoustics, Speech and Signal Processing 2008 (ICASSP 2008) [54]. Other sections of text wre reprinted from “A Concatenated Model for Video Frame Interpolation,” which has been published in IEEE 13th Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop 2009 (DSP/SPE 2009) [55]. The dissertation author was the primary researcher and author, and the listed co-author in the publication supervised and directed the research and also contributed to this chapter.

5. MOTION MODELS USED FOR VIDEO COMPRESSION

In this chapter, we consider the video coding application and the use of our new models in that domain. In video compression, the motion is assumed to be translational. Our method uses a higher order model to characterize motion, which allows it to achieve higher accuracy when compared with conventional block matching algorithm. The chapter starts with the introduction of the video compression standard. This is followed by a discussion of the difficulty in detecting reliable motion vectors from compressed video sequences. The overall system framework is described and experimental results are provided. Some challenging issues are discussed at the end.

5.1 Overview of the H.264 Video Coding Standard

Video compression has been widely used in applications such as digital television, DVD-Video, mobile TV, video conferencing and internet video streaming. H.264/AVC is the current industry video coding standard for video compression. It builds on the concepts of earlier standards such as MPEG-2, MPEG-4 Visual, and H.263.It offers better compression efficiency (i.e. better-quality compressed video) and greater flexibility in compressing, transmitting, and storing video.

A basic flowchart of the H.264 video coding process is shown in Figure 5.1. When encoding a video sequence, each frame is partitioned into macroblocks, which are defined as 16×16 blocks. Each macroblock is using either intra prediction or inter prediction so that a prediction block is formed and is subtracted from the original macroblock to form the residual frame. Quantization and entropy encoding are performed on the residual frames to form bitstreams. Then they are transmitted over networks. At the decoder side, the bitstreams are first entropy decoded, rescaled, and inverse transformed to form the uncompressed residual frame. By decoding the motion vectors, predicted frames are reconstructed

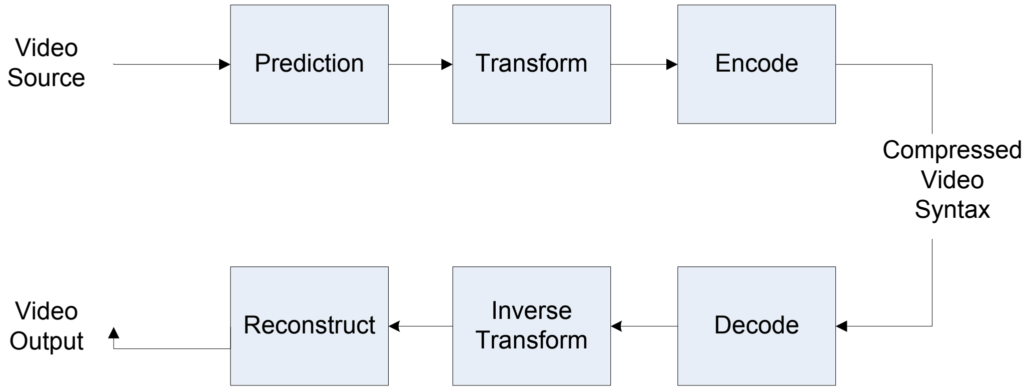


Fig. 5.1. H.264 overall video coding scheme

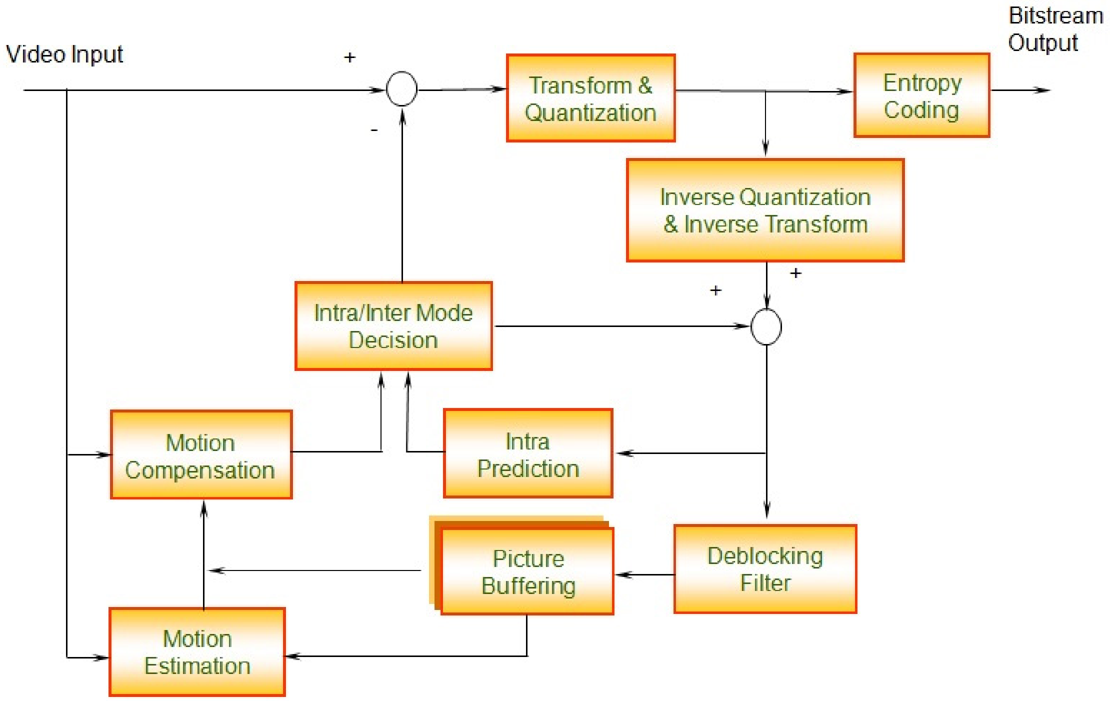


Fig. 5.2. Standard H.264/AVC encoder diagram

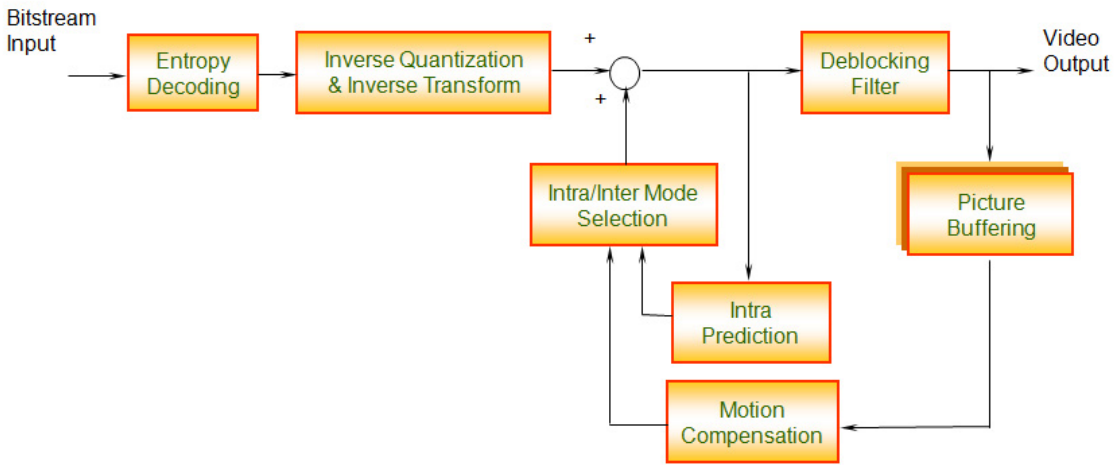


Fig. 5.3. Standard H.264/AVC decoder diagram

at the decoder side. Adding the predicted frame to the residual frame results in the reconstructed frame. The encoder diagram and decoder diagram are illustrated in Figures 5.2 and 5.3 respectively.

The major improvements associated with H.264 are using variable block size, performing integer block transforms, and improving the in-loop deblocking filter. It achieves much better coding performance than older standards such as MPEG4 and H.263L.

5.2 Issues in Compressed Video

The difference between the uncompressed video and compressed video is that after the compression and decompression processes, which involve block transformation, the decompressed video has blocking artifacts and other distortions. The motion vectors that are generated in the encoding process are not explicitly optimized for accuracy but for achieving minimum mean square errors. At the decoder side, the decoder uses the motion vectors generated by the encoder to reconstructed the original frames. The difference is dependent on the bit budget - the fewer the number of bits used to compress the video sequences, the worse the reconstructed video quality.

The scenario we want to handle now is not to spatially enhance the resolution, but to achieve coding efficiency. We intentionally downsample the HR sequence and thus we can transmit it over the network using fewer bits. At the decoder side, we perform the spatial enhancement and enlarge the video frames. In this way, we can apply our method in a video compression scenario and provide scalability as well.

5.3 Overview of the New System

The proposed method employs H.264/AVC as an autonomous component and is thus completely compatible with the standard. Our overall coding system consists of three stages: pre-processing, compression/decompression, and post-processing, which are illustrated in Figure 5.4. The pre-processing stage conditions the sequence for efficient encoding at the second stage in the whole system, which is illustrated in Figure 5.5. The input to the pre-processing stage is the original $M \times N$ video sequence. Every Lth frame is retained in its original spatial resolution while the other frames are decimated to $M/2 \times N/2$, resulting in the LL (low-low) subband frames. A 21-tap lowpass half-band filter is used before the downsampling. The output is a mix of low spatial resolution and high resolution frames. Not all the frames need to be coded at full resolution because the proposed method can

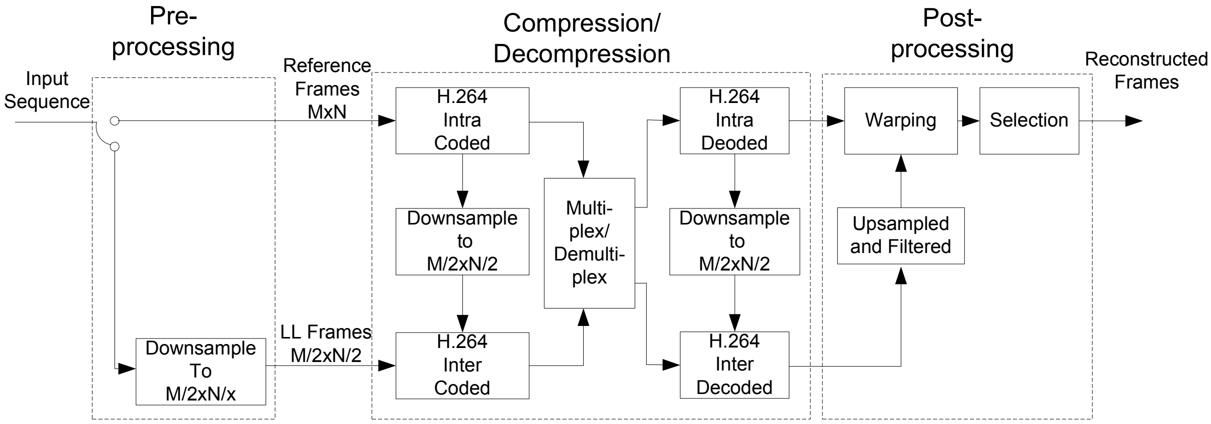


Fig. 5.4. System diagram

convert the decimated frames to full resolution. Lower resolution video sequences require fewer bits to code which results in higher compression. But we also want to retain the necessary high spatial information to enhance interpolated frames in the post-processing stage. The core of the compression/decompression stage is H.264/AVC. At the encoder, the high resolution reference frame is intra-coded and downsampled to $M/2 \times N/2$ and stored in the frame buffer so that it is available for inter-coding. The intra-coding and inter-coding employ H.264/AVC because of its efficient performance. However, because the spatial size of the LL subband is one-fourth that of the original, we can reduce the bit rate relative to the coded full band sequence. The total bit rate is the sum of the intra-coded frames and the inter-coded subband frames. At the decoder, the full resolution frames are first decoded using H.264/AVC and downsampled to half resolution in both horizontal and vertical dimensions. Being stored in the frame buffer, they can be used to decode subsequent LL subband frames. At this point, we have decoded both the high resolution reference frame and low resolution LL frames.

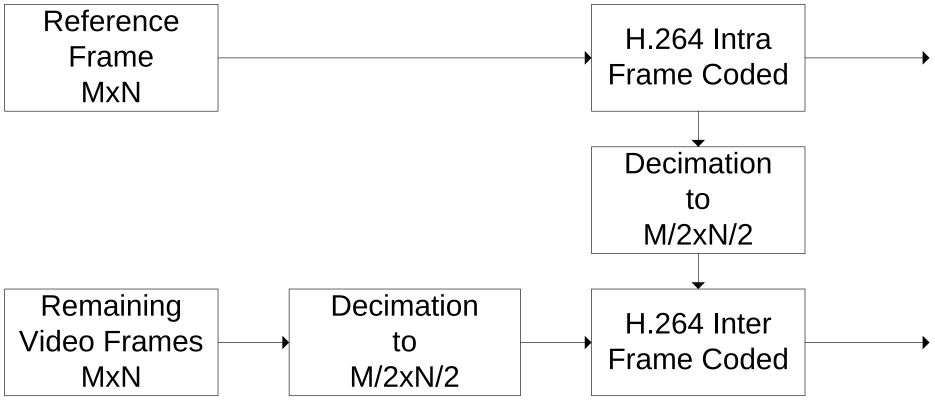


Fig. 5.5. Simplified encoder diagram

In the post-processing stage, which is illustrated in Figure 5.6, we first upsample the LL subband frames to full resolution using the same lowpass filter. This is followed by applying the modified warping algorithm to the associated reference frame to produce the high quality reconstructed frame. Thus we maintain high frequency memory over an L-frame interval of the sequence.

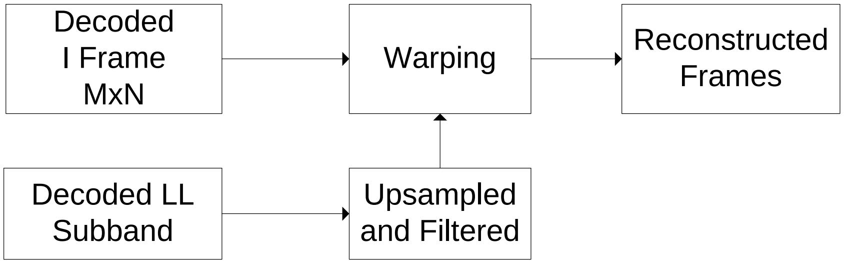


Fig. 5.6. Post-processing diagram

5.4 Experimental Results

We used JM 11.0 reference software and evaluated the algorithm on a variety of standard video test sequences. We setup two sets of experiments. In the first we compare the new coding scheme with H.264/AVC. To have a fair comparison, we used the same GOP structure to encode the sequences for both the conventional H.264/AVC encoder and the proposed encoder. That is, for the H.264/AVC encoder, we encoded every Lth frame as an I frame and the rest of the frames as P or B frames. For our proposed method, every Lth frame is designated as a high resolution reference frame and is intra-coded. For the frames in between, we first downsample them by a factor of two in both horizontal and vertical directions to get the low resolution subband frames and inter-coded them. The total bit rate for the proposed method is the sum of the bit streams from the full resolution reference frames and the quarter resolution subband frames. The PSNR we report is the average over the whole sequence. When implementing, we set the GOP structure as IPPP and every $20th$ frame is coded as a reference or I frame. The remaining frames are coded as P frames. This is because the baseline profile doesn't support B frames. The low to moderate motion sequences we tested include Akiyo, News, Salesman, Foreman and Carphone.

Compared to the H.264/AVC coder, the proposed method produces higher PSNRs at low

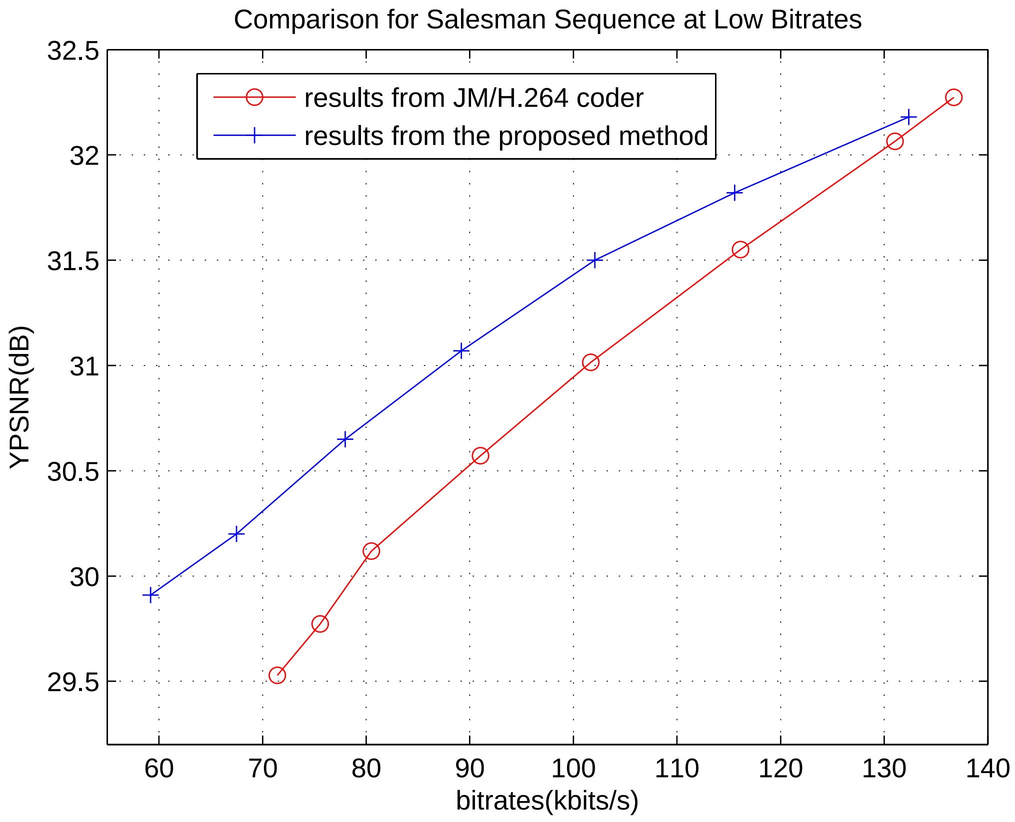
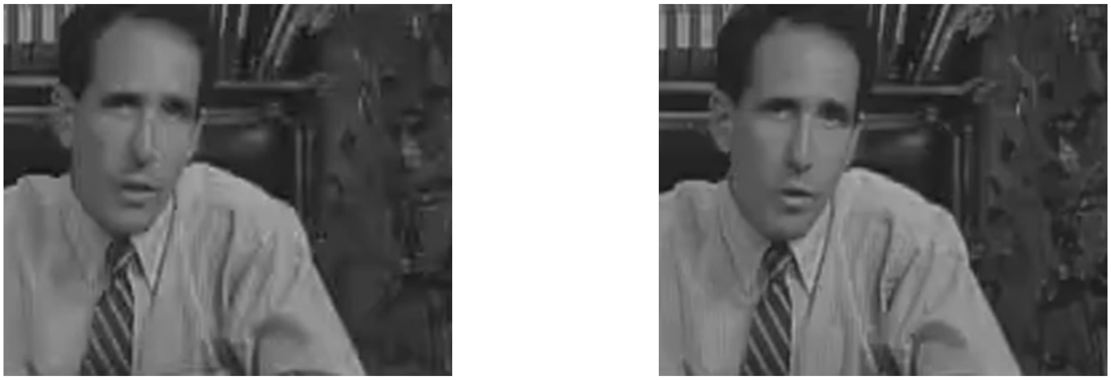


Fig. 5.7. Comparison for Salesman sequence at low bitrates

bit rates ranging from 60Kbits/s to 135Kbits/s depending on the sequence. However, as the bit rates go up, the PSNR differences diminish and eventually PSNRs for the decoded sequences from the H.264/AVC exceed the ones from the proposed method as illustrated in Figure 5.7 for the Salesman sequence. The PSNR performance of the H.264/AVC coder improves significantly at higher bit rates as does the spatial resolution.

However, what is evident in the comparison is that the subjective quality of the proposed approach is much higher at low to moderate bit rates. The frames look sharper and more natural while the H.264/AVC results tend to have block artifacts and appear blurry by comparison. Figure 5.8 provides a subjective comparison of the performance of the H.264/AVC coder and the proposed coding scheme. The frame shows the 3rd frame of the Salesman sequence coded at 89 kbits/sec. The left image is the decoded frame using H.264/AVC. The right one is the frame using the proposed method. Both results are good. The differences are small, but visible. As we can see, around the salesman's nose and face, the H.264 coder produces subtle blocking artifacts, which are not present with the new method. Also around the books on the bookshelf in the background, our resultant frame contains higher definition.

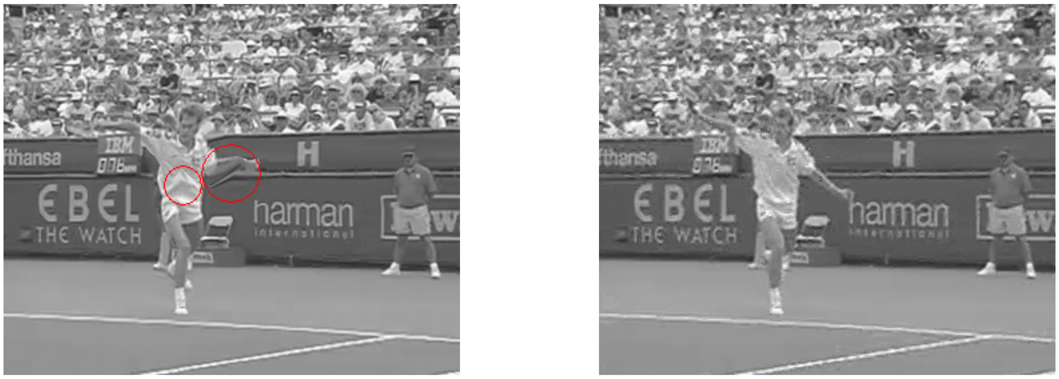


(a) Frame 3 coded with H.264/AVC

(b) Frame 3 coded with the warping method

Fig. 5.8. Comparison of the subjective quality of H.264/AVC and the warping method for frame 3 of the Salesman sequence at 89 kbits/sec

The second experiment is to compare the composite algorithm with the full band warping algorithm (a.k.a. forward warping) on videos with medium to high motion. For the full band warping algorithm, the three stages are similar. The pre-processing stage and the coding stage remain the same. The difference is that in the post-processing stage, the selection algorithm uses the forward warping method exclusively. We tested higher motion sequences such as *Stefan* and *Table Tennis*, which have moderate to high motion. The GOP structure uses every 5th frame as the reference frame. The rest of the frames are coded as P frames. As reported by Chen et al. [56], the forward warping algorithm will fail on sequences with moderate to high motion because the motion can no longer be accurately represented with translational pixels. Depending on the nature of the motion, as shown in Figure 5.9 where it depicts the 28th frame in the Stefan sequence, the original forward warping algorithm produces false motion fields, leading to a geometrically distorted area around the tennis player's left arm. The composite algorithm addresses this problem to a large extent. The composite algorithm does blur the output frame slightly because it tends to decrease the difference between the warped frame and the decoded blurry low resolution subband. But the tradeoff results in improved subjective quality overall.



(a) The forward warping method

(b) The composite algorithm

Fig. 5.9. Comparison of the subjective quality of the forward warping and the composite algorithm for coded frame 28 of the Stefan sequence.

5.5 Discussion

There are several other factors that affect the overall performance, such as rate control, bit allocation, and frequency of the reference frames. Rate control is used to make sure the target bitrates are met. It provides a fair comparison between the results from standard coding and the ones from our method. Bit allocation is used to determine how many bits are spent on the reference frames and low resolution frames. Different allocations lead to different levels of quality of the produced video because in the video coding, the quality of P/B frames are highly dependent on the quality of the I frames. Higher quality I frames lead to higher quality P/B frames even though the bits spent on the P/B frames may be lower than the other case when I frames are poor and more bits are spent on P/B frames. How often we choose a reference frame also has an impact on the results because the farther the frame is away from the reference frame, the worse the quality of the warped frame. A potentially attraction feature of this new approach is that reference frames might be recycled in steady state. Originally designed with the intent of coding lecture video, it is imagined that a base set of reference frames can be employed after an initialization period thereby avoid having to transmit those frames periodically. This we feel is a promising area for further exploration.

5.6 Summary

Improving the spatial quality of video coded at low bit rates is a problem of general interest. Toward this end, we proposed a combined forward warping, backward warping method to improve the resolution of video encoded with H.264/AVC. The scheme attempts to capture long-term spatial detail by warping high resolution reference frames in accordance with displacement vectors derived from decoded low resolution frames. At low bit rates, the proposed method can achieve better PSNRs and better subjective quality than conventional H.264/AVC for sequences with low to moderate motion. However, as the bit rates go up, the H.264/AVC coder has more bits to spent on the high frequency information and thus the resulting frames don't have obvious block artifacts and tend to do better in preserving spatial detail. Consequently the gap between the proposed method and H.264/AVC coder decreases with increasing bit rate.

5.7 Acknowledgments

Part of the text in this chapter is a reprint from "An Approach to Enhanced Definition Video Coding Using Adaptive Warping," which has been published in the Proceedings of Visual Communications and Image Processing 2009 [57]. The dissertation author was the

primary researcher and author, and the listed co-authors in the publication supervised and directed the research and also contributed to this chapter.

6. TEMPORAL INTERPOLATION

Video frame temporal interpolation has attracted much attention over the years, largely because of commercial frame rate up-conversion applications, such as 24p to NTSC and 24p to PAL. In this chapter, a new motion compensated frame interpolation method is proposed based on the reliability of motion vectors determined by the block residual energy. Additional motion re-estimation is applied to those blocks where unreliable motion vectors are detected. The algorithm presented in this chapter builds on the methods introduced in last three chapters, resulting in a high performance method with only modest computational complexity. The experimental results show that the new method can improve the visual quality of the interpolated frames where competing methods fail.

6.1 Introduction

For many video coding applications, systems seek to operate at reduced rates, in which case temporal resolution is routinely scaled back to better preserve the spatial quality of the sequence. In order to play back the video, frame-rate up-conversion is performed to restore the missing frames. For LCD display applications, high frame rate video is desired in order to reduce blurring, particularly for fast motion video. Quality is often improved by up-converting the frame rate of standard video captured at 30 Hz by a factor of two or more. For media broadcast of movies, frame-rate up-conversion is critical to accommodate the frame rate difference of the industry standards. The movie industry typically operates with a 24 frame/second capture rate, while media broadcasts employ a 30 Hz standard. Indeed, there are many applications in which frame-rate up-conversion is necessary and high quality is important.

A number of approaches to frame-rate up-conversion have been considered and reported in the literature. Among the simplest approaches are frame replication and linear interpolation [58]. While these methods are attractive from a low complexity standpoint, they often result in visual distortions, primarily in the form of blurring. An approach that generally improves quality relative to frame replication and temporal linear interpolation is to use motion compensated frame interpolation [59–61]. Best results are achieved when the motion vectors have high accuracy. Most methods can be categorized into one of two different classes: the first employing existing motion vectors which, for example, might be obtained from the decoder; the second involving re-estimating the motion vectors for the motion compensated interpolation. It has been observed that motion vectors employed in some of the coders (like H.264/AVC) do not perform as well, owing to the block matching algorithms used to derive them. A more fundamental deficiency associated with block matching motion estimation is the limitation in representing complex motion such as rotation and zooming, as opposed to simple translational motion. Performance improvement can be achieved by adopting strategies that identify errant motion vectors and replace them with recomputed estimates derived from surrounding vectors. A number of authors have reported on strategies of this type [62–64].

In this chapter, a new approach to this problem is considered and evaluated. It sits between the two classes mentioned above and involves motion re-estimation performed selectively according to the reliability of the motion vectors delivered from the decoder. The motion estimation stage shares much in common with the method of Frakes et al. which was used to interpolate between slices of magnetic resonance images [30, 65]. The proposed approach is different in the method employed to select the vectors and refine accuracy. In particular, additional motion re-estimation is applied to those blocks where unreliable motion vectors are detected. At some point it is necessary to perform the motion estimation again to get the true motion vectors because of the limitation of the conventional block-base motion estimation performed in the encoder.

This chapter is organized as follows. In Section 6.2, the overall system, which includes four stages, is proposed. Each of the elements is described in detail. The experimental setup and results are reported in Section 6.3, followed by conclusions provided in Section 6.4.

6.2 The General System

Achieving good motion estimation is the key to obtaining sharp temporally interpolated frames. If one employs a simple frame averaging approach to derive a middle frame between two consecutive frames, the resulting frame is generally blurry. Employing motion vectors in the process clearly yields better results. However, motion estimation is very computationally intensive. Thus, there is motivation to use motion vectors computed and transmitted by the encoder. In the meantime, not all the motion vectors transmitted by the encoder can be used for motion compensated interpolation because all may not correspond to true motion. A critical part of the overall system is to detect unreliable motion vectors and to correct them. The overall system consists of four parts: motion reliability checking, small block merging, motion vector re-estimation, and frame interpolation as shown in Fig. 6.1. The inputs to the system are the previous reconstructed frame X_1, the current reconstructed frame X_2, the current residual frame, and the motion vectors, all of which are from the decoder. The goal of the motion reliability check is to identify the unreliable motion vectors. The goal of the small block merging is to prepare for the next stage so that it can perform the motion estimation using a larger area. The motion re-estimation stage employs the motion estimation algorithm presented in Chapters 2 and 3. More details will be discussed in Section 6.2.3. The last stage of the whole process in frame interpolation uses the motion vectors derived in the previous stages. Two challenges are avoiding blocking artifacts and detecting occlusion. Overlapped motion compensation [66] is used to alleviate the blocking artifacts while uni-directional motion compensation, either the forward or backward, which ever gives the better boundary absolute difference, is adopted to handle occlusions.

6.2.1 Motion Reliability Check

The motion vectors delivered from the decoder are not always sufficiently accurate. This can happen when background is uncovered, objects within one block move in different directions, or the motion is more complex than translational. Motion vector inaccuracies

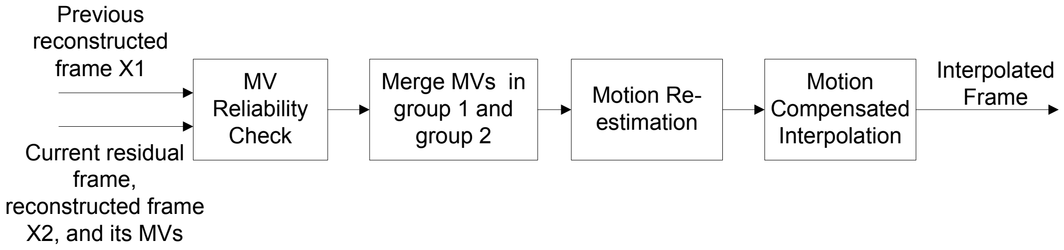

Previous reconstructed frame X1
Current residual frame, reconstructed frame X2, and its MVs
MV Reliability Check
Merge MVs in group 1 and group 2
Motion Re-estimation
Motion Compensated Interpolation
Interpolated Frame

Fig. 6.1. System diagram

may also be attributed to the use of a selection criterion that minimizes residual energy. To improve performance, we need to have measures to determine whether the motion vectors from the decoder are reliable. Toward this end, we use the residual energy [62].

Motion vectors are parsed into three groups in accordance with the residual energy. An 8×8 block size is used. If the residual energy in the block is above a preset threshold, the block is labeled as group 1, indicating a low accuracy motion vector. Intra-coded blocks are also put into group 1. If the block residual energy is below the threshold, but one of its eight surrounding blocks is in group 1, the block is categorized as group 2, which denotes the possibility of it being unreliable. The remaining blocks are classified as group 3, indicating that they are reliable. After this stage, a block reliability map (BRM) is formed representing the unreliable, questionable, and reliable groups. This process is illustrated in Fig. 6.2

6.2.2 Small Block Merging

A problem can occur when there are repeated patterns, as observed by other investigators [63]. The temporal interpolated frame tends to introduce broken edges in objects. To overcome this shortcoming, small blocks containing errant motion vectors are merged to form a larger block so that object structure and edge continuity are maintained. Blocks are merged according to the block reliability map mentioned in the previous section. Only group 1 or 2 blocks are merged. Blocks in group 3 are kept the same. Some typical cases are shown in Fig. 6.3 to illustrate how the merging process works. Fig. 6.3(a) demonstrates

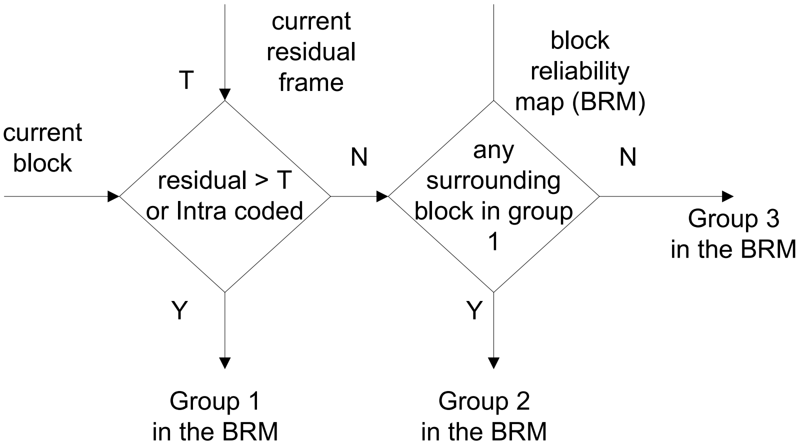


Fig. 6.2. Flow diagram for group calculation for the block reliable map.

the case where two horizontal blocks from either group 1 or group 2 are grouped into one block of size 8×16. Two vertical blocks in either group 1 or group 2 are merged to a block of size 16×8 as shown in Fig. 6.3(b). If three out of four blocks of size 8×8 in an isolated 32×32 block are in group 1 or 2, we merge them into a block of size 16×16 as illustrated in Fig. 6.3(c). One special case to note is when two blocks in group 1 or 2 occur in a diagonal configuration. In such a case, they are not merged, which is shown in Fig. 6.3(d). There are many other cases where similar principles can be applied. The largest block size we allow is 32×32, which was determined empirically to yield the best results.

6.2.3 Adaptive Motion Field Re-estimation

For blocks where there is large residual energy, the motion vectors obtained from the decoder are typically not reliable. One can address this issue by using surrounding motion vectors to correct the unreliable motion vectors [63]. However, if the areas involve complex motion such as rotation or camera zooming, using surrounding motion vectors does not improve performance. Consequently the approach taken in this chapter is to re-estimate the motion vectors.

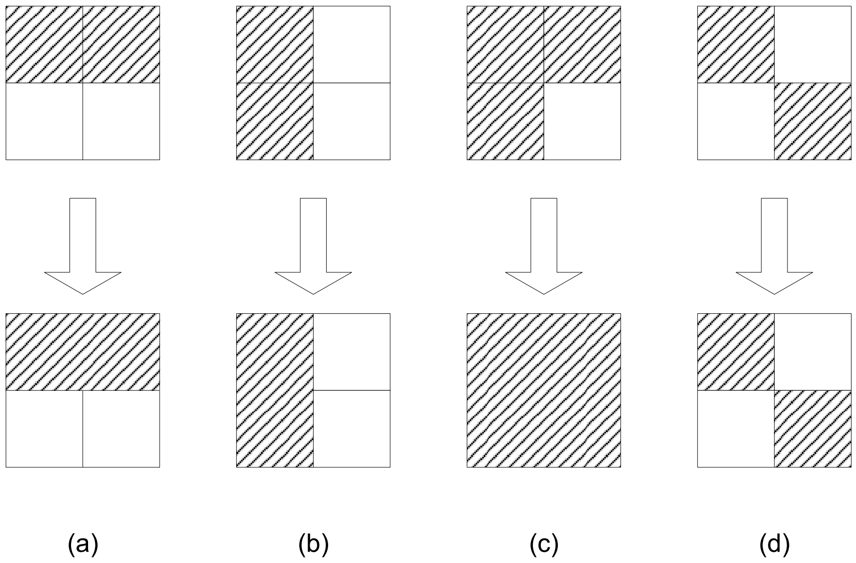


Fig. 6.3. Examples of typical blocks encountered in the merging process. (a) Merging two horizontal subblocks. (b) Merging two vertical subblocks. (c) Merging three subblocks. (d) Processing unreliable or questionable subblocks.

The motion estimation strategy used in our system incorporates the motion estimation methods described in Chapter 3. In particular, we again use the bilinear motion model where displacements are a weighted sum of four basis vectors as described by Equation (3.13) and (3.15), and when the weighting parameters are obtained through Equation (3.18). The method is applied in a quadtree framework where regions of high motion are successively split into subblocks based on the computed mean square error.

6.2.4 Motion Compensated Interpolation

Bilateral Interpolation

After all the motion vectors are derived, motion compensated interpolation is applied to create the frames in the middle. Given two frames X_1, X_2 and a motion vector $\vec{mv}$

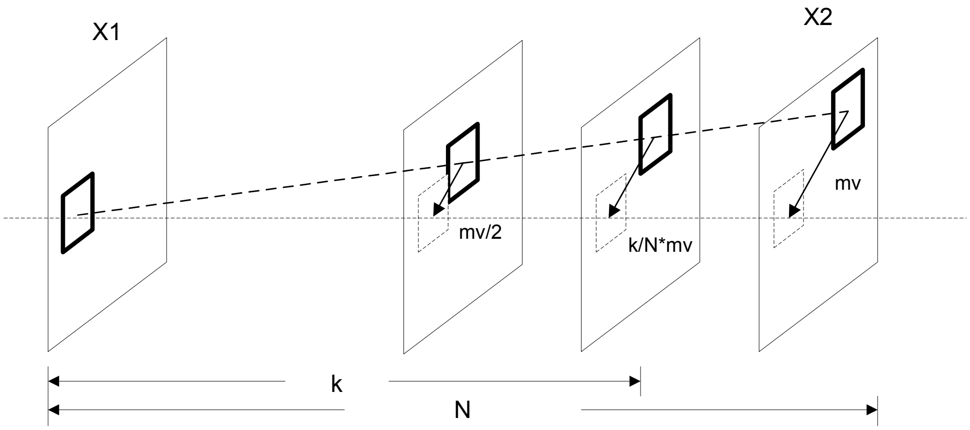


Fig. 6.4. Motion vectors for motion compensated interpolation.

(pointing from X_2 to X_1 as shown in Fig. 6.4), the frame in between, denoted as $X_m[\vec{x}, k]$, is given by

$$X_m[\vec{x}, k] = \frac{N-k}{N} X_1[\vec{x} + \frac{k}{N}\vec{mv}] + \frac{k}{N} X_2[\vec{x} - \frac{N-k}{N}\vec{mv}], \tag{6.1}$$

where $k = 1, 2, \ldots, N-1$. Here N denotes the distance between two consecutive frames from the original sequence after interpolation. In other words, we want to temporally interpolate between two frames by adding frames in between. In our notation, k denotes the specific position of the interpolated frame. If $N = 2$ and $k = 1$, then Eq. (6.1) reduces to

$$X_m[\vec{x}] = \frac{1}{2} X_1[\vec{x} + \frac{1}{2}\vec{mv}] + \frac{1}{2} X_2[\vec{x} - \frac{1}{2}\vec{mv}], \tag{6.2}$$

which means only one frame is to be created after each frame. For simplicity of explanation, we assume $N = 2$ and $k = 1$ in the following sections.

Uni-directional Motion Compensated Interpolation Method

In areas where occlusion occurs, a uni-directional motion compensated interpolation method is used to produce smooth interpolated frames. To decide whether there is an occlusion, the difference of the forward interpolation and backward interpolation "DIF(block)" is computed on a block-by-block basis where

$$DIF(block) = \sum_{\vec{x} \in block} |X_1[\vec{x} + \frac{1}{2}\vec{mv}] - X_2[\vec{x} - \frac{1}{2}\vec{mv}]|.$$

The assumption is that if there is an occlusion, there will be a large difference between the frame created directly from the forward interpolation and the frame created from backward interpolation. If the difference exceeds a threshold, that block is created using either forward interpolation or backward interpolation, which ever gives the smaller boundary absolute difference (BAD). The BAD is denoted as the difference between the resulting boundary pixels of the motion compensated block and the surrounding pixels as can been seen in Fig. 6.5. The BAD is the sum of the absolute difference between the red pixels and blue pixels when they are available. Thus the resulting block looks smoother with regard to the surrounding blocks.

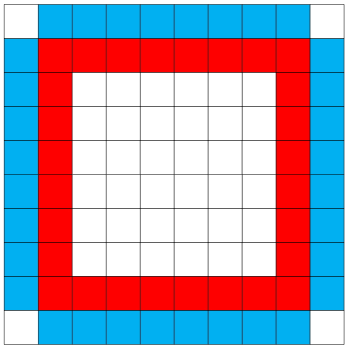


Fig. 6.5. Boundary absolute difference illustration.

Overlapped Boundary Motion Compensation

Generally speaking, because the motion compensated interpolation is conducted on a block-by-block basis, blocking artifacts can be visible, as noted by many authors. To alleviate the blocking artifacts, overlapped block motion compensation (OBMC) [66–69] was proposed in the literature with different variations. It is reported in [64] that OBMC can lead to blurring if it is performed on all blocks. Here a selective OBMC is used on blocks where the BAD is above a preset threshold. Only the boundary pixels will be replaced for those blocks.

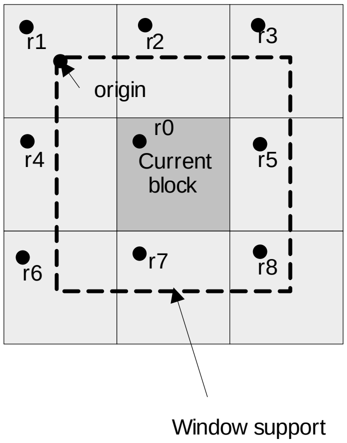


Fig. 6.6. Window support and qualified pixel locations.

OBMC predicts a pixel value using a linear combination of the estimates given by motion vectors of a block and its neighboring blocks. $B = \{-1, 0, 1\} \times \{-1, 0, 1\}$ is used to denote the neighboring block of the current block. Assume that $B(-1, -1)$ refers to the upper left subblock of the current block. A symmetric window $w(\vec{x})$ of size $2N \times 2N$, whose center is the center of the current block, is used to cover a wider region. N denotes the height and width of the block and $\vec{x} = (s, t)$ is the position of each pixel in the current block. Each pixel position becomes $\vec{x}' = (s + \frac{N}{2}, t + \frac{N}{2})$ because the upper left position of the window is set as the origin, where (s, t) are the coordinates according to the original origin. Fig. 6.6 shows a graphical illustration. For each pixel in the current block, there are 8 corresponding pixels in the neighboring block. Only 3 of these 8 pixels are covered within the window. The pixel value is predicted as a linear combination of the four compensated values derived by their corresponding motion vectors with different coefficients, as defined by the equation

$$X_m[\vec{x}] = \sum_{i=0}^{3} w_i(\vec{x})(\frac{1}{2}X_1[\vec{x} + \frac{1}{2}\vec{mv}_i] + \frac{1}{2}X_2[\vec{x} - \frac{1}{2}\vec{mv}_i]). \tag{6.3}$$

The parameters w_i and $\vec{mv}_i$ $(i = 1, 2, 3)$ are the coefficient weights and motion vectors for the other 3 pixels in the corresponding blocks mentioned above. The coefficient weights are obtained from the equations

$$w(\vec{x}) = w'(s)w'(t) \tag{6.4}$$

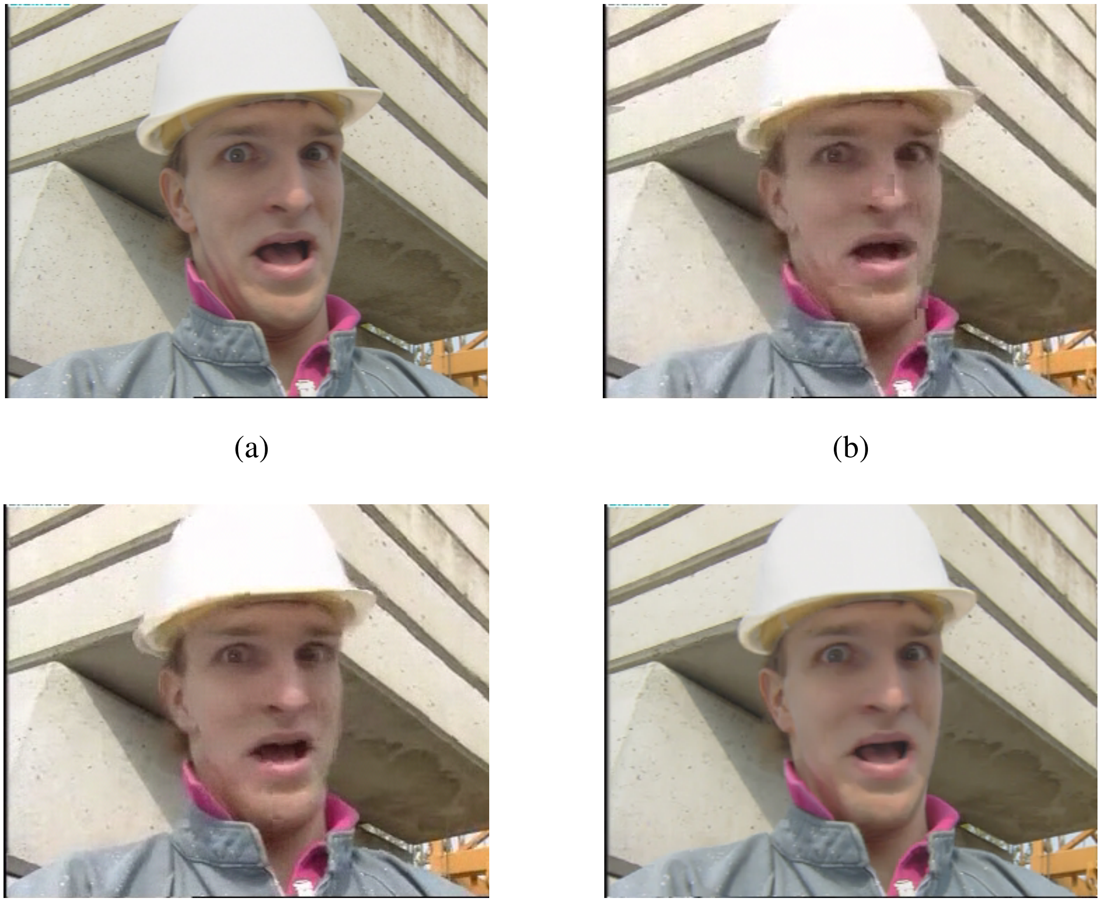
$$w'(s) = \begin{cases} \frac{1}{N}(s + 0.5) & s = 0, \ldots, N-1 \\ \frac{1}{N}(2N - s - 0.5) & s = N, \ldots, 2N-1. \end{cases} \tag{6.5}$$

Similarly, $w'(t)$ is obtained in the same way where here s and t are the relative coordinates with respect to the origin of the $2N \times 2N$ window.

6.3 Experimental Results

We used JM 11.0 and tested the algorithm on three sets of video sequences. The first set consists of “talking head” videos and includes Akiyo, News, Salesman, Foreman, and Carphone sequences. These sequences have low to moderate motion. The second set contains sequences with high texture and includes Flower Garden, Tempete, Bus, Mobile, and Calendar. The third set is characterized by having large motion and includes Football, Stefan, and Table Tennis. To simplify the test, we use fixed QP for the sequences. The GOP structure is IPPP with every 15th frame as an I frame. Transform 8×8=2 mode is used, which only considers block sizes of 8×8. The search range for all the sequences is 16. Every odd frame is coded and every even frame is skipped. The video sequences are coded at 15fps. We compared our proposed method against direct motion compensation (DMCI) and correlation-based MV Selection [70].

The subjective results from the Foreman sequence coded at 384 kbps are shown in Fig. 6.7. As can be seen, the result using direct motion compensation leads to visible blocking artifacts. The correlation-based MV selection method shown in Fig. 6.7(c) alleviates the artifacts to a certain extend but the result contains noticeable distortion. The proposed method overall performs better than the ones tested. This improvement is illustrated in Fig. 6.7(d), where it is evident that the distortions have been suppressed. The YPSNR is also much higher than both competing methods.



(a)

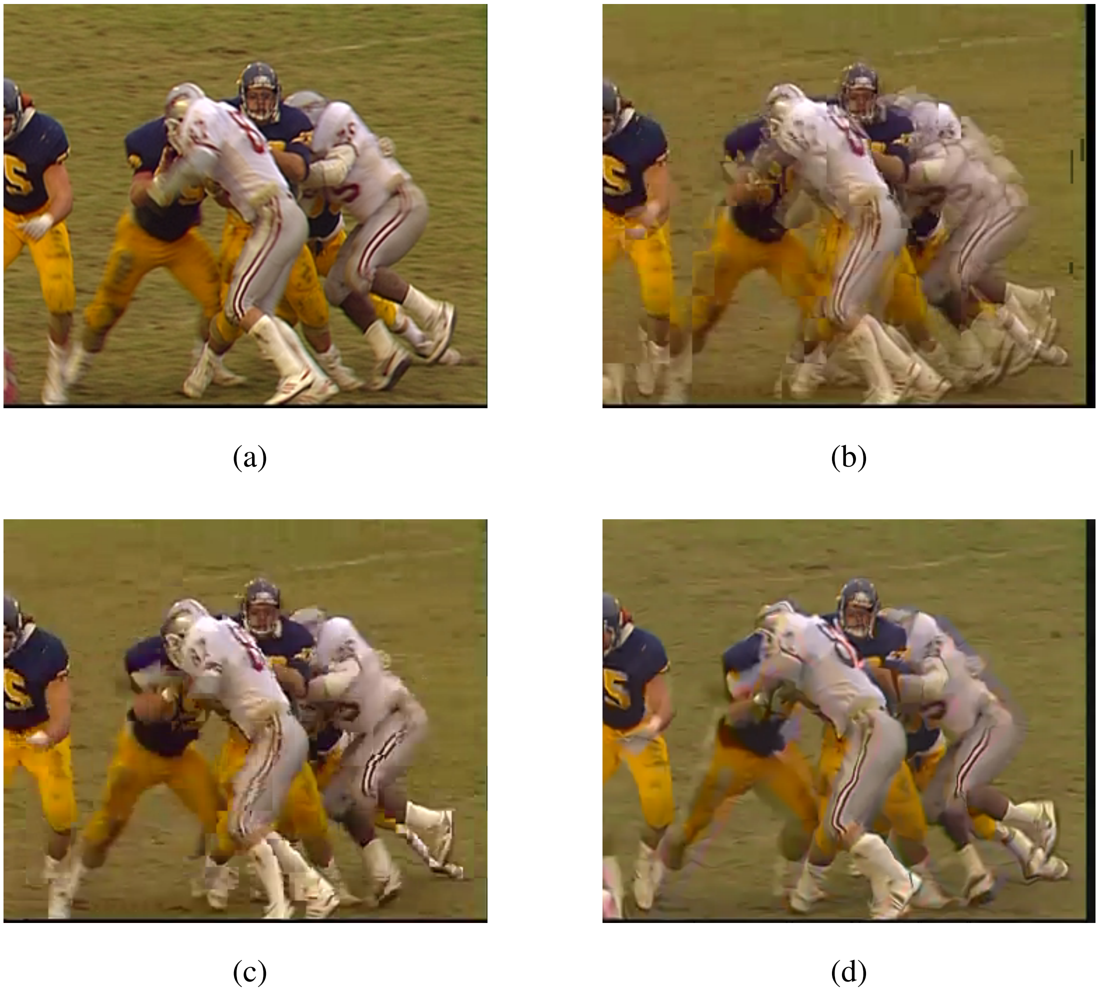
(b)

(c)

(d)

Fig. 6.7. The interpolated results from Foreman sequence (a) Original frame 96 (b) Result using direct motion compensation (YPSNR: 20.54dB) (c) Result from correlation-based MV Selection (YPSNR: 20.48dB) (d) Result from the proposed method with adaptive motion re-estimation (YPSNR: 24.19dB)

Fig. 6.8 shows the results from the Football sequence coded at 512 kbps. This sequence has large motion and is generally challenging for motion compensated interpolation. The



(a) (b)

(c) (d)

Fig. 6.8. The interpolated results from the Football sequence (a) Original frame 52 (b) Result using direct motion compensation (YPSNR: 20.01dB) (c) Result from the correlation-based MV selection (YPSNR: 20.17dB) (d) Result from the proposed method with adaptive motion re-estimation (YPSNR: 20.13dB)

blocking artifacts are evident in Fig. 6.8(b). The correlation-based MV selection method reduces most blocking artifacts. However, around the arms of the two football players in the center, the blocking artifacts are still obvious. Compared to the correlation-based MV selection method, the proposed method produces smoother result. The PSNR for the proposed method is about the same as the correlation-based MV selection method and the subjective quality is comparable. The point here is that this algorithm works best for low to medium motion situations but performs on par with other methods when motion is high.

6.4 Conclusion

In this chapter, we address temporal interpolation, complementing the spatial interpolation methods introduced in the previous chapters. In particular, a motion compensated frame interpolation method is presented that builds on the bilinear motion model and motion estimation method described in Chapter 3. Unreliable motion vectors are first detected, and then motion estimation is performed in those areas. Occlusions are handled by choosing either forward interpolation or backward interpolation, which ever produces the better Boundary Absolute Difference on the surrounding blocks. A selective overlapped boundary motion compensation is applied to reduce the blocking artifacts. Experimental results show that the proposed method performs better on video sequences containing low to medium amounts of motion and is comparable to the correlation-based MV selection method for high motion sequences.

6.5 Acknowledgments

Part of the text of this chapter is a reprint from “Adaptive Motion Estimation Using Warping for Video Frame Rate Up-conversion,” which has been published in the Proceedings of Visual Information Processing and Communication 2010 [71]. The dissertation author was the primary researcher and author, and the listed co-authors in the publication supervised and directed the research and also contributed to this chapter.

7. CONTRIBUTIONS AND FUTURE WORK

Advances in microelectronics and the digital infrastructure within society have increased the demand for high performances video processing systems. At the core of most modern video processing algorithms is a generic motion model to describe the complex nature of video signals. In this document, a general framework is proposed to handle several different processing tasks, such as spatial interpolation, video coding, and temporal interpolation on uncompressed and compressed video sequences. The original application motivating the research detailed in this thesis is compression and display of “talking head” video sequences. Video of this type is dominant in the distance learning area. As the work matured, more complex sequences were considered along with the issues involved in manipulating complex motion. The contributions of this dissertation are summarized in the succeeding sections.

7.1 Contributions

A major contribution of this thesis pertains to development of high performance motion models. These are discussed in Chapter 4 where a hybrid warping scheme is proposed to handle occlusions and high motion videos. The new models build on Adaptive Control Grid Interpolation (ACGI) and bidirectional warping to fuse high frequency components to either the preceding low resolution frames or the succeeding frames. The critical part in the framework is the motion estimation stage, which is based on modified adaptive control grid interpolation. Investigations supported by experimental results show the failure mode of ACGI, which can introduce motion vector estimates that are unreliable. To compensate the shortcoming, unreliable motion vectors are first detected, and then their associated pixels are removed by choosing the best pixels among the forward warped frame, backward warped frame, and spatially interpolated frame. This new model was shown to improve performance significantly.

A number of other contributions can be cited that pertain to video frame interpolation, video coding, and video temporal interpolation applications.

- Video frame spatial interpolation

  Numerous video/image interpolation methods have been investigated and developed in the literature. Most of these algorithms are focusing on the spatial property of the images and video frames. The application we consider is mainly interpolation of “talking head” videos where the background is static, and the head movement in the foreground is constrained. The adaptive motion model can re-align itself by analyzing the video sequences characteristics and achieve much higher subjective quality than competing methods. For sequences with large motion, the motion selection process becomes more conservative, thus the subjective quality is similar to the direct spatially interpolated results.

- Video coding

  The nature of the proposed motion model makes it easy to apply to video coding applications. The model employs several parameters to represent displacement and warping within a region. The accuracy of this representation is similar to pixel-based optical flow method. By downsampling the sequence at the encoder side and interpolating it at the decoder side, performance improvements were achieved. A distinguishing feature of this work is in the post-processing stage where the motion estimation is performed. This is different from the SVC (Scalable Video Coding) framework where the major change is made at the encoder side.

- Video frame temporal interpolation

  The ability of the new models to obtain an accurate motion field made them effective for video frame temporal interpolation since accurate motion estimation is crucial. The overall performance of the new method was shown to be better than the correlation-based MV selection method, which is considered one of the best motion compensated frame interpolation methods for video sequences with low to medium motion. For video sequences with high motion, it also achieved comparable results.

There is a tradeoff between the frequency of the high resolution frames and the robustness of the motion estimation, which is also content-dependent. If a video sequence contains low motion, then inserting high resolution frames less often is recommended, where the resultant frames still look sharp and correct. On the other side, a high motion sequence requires higher rate of high resolution reference frames because large motion will violate the fundamental limitation of the optical flow-based motion estimation; thus the target frame can't be too far away from the reference frames. Lower rate of high resolution reference frames leads to higher compression ratio in the video compression application because I frames normally consume most of the bitrates. This tradeoff sets the limitation of what types of the input videos can be in the spatial interpolation area and it determines the highest frame rate the algorithm can achieve given a fixed input video source in the video frame temporal interpolation application.

7.2 Future Work

7.2.1 Open Issues of Robustness

The two-dimensional projection of the three-dimensional motion vector makes the motion estimation process an ill-posed problem because many different movements can lead to the same observation. For real video sequences, motion is a combination of different types of movement that might not be linear. No ground truth is available for those sequences with complicate motion. Synthetic sequences might mimic several types of motion, however, how it relates to real sequences is still an open problem. Two scenarios are deserving of special attention. First is the case of covered and uncovered objects or background leaving or entering the scene. Second is the case of non-rigid objects. Although by using bidirectional warping, the missing pixels can be found either from the preceding frame or the succeeding frame, we believe having a separate occlusion detector that is more complex could help improve the results.

7.2.2 Reference Frame Recycling

As mentioned in the introductory sections of this thesis, distance learning is an emerging area. Efficient coding of lecture head and shoulders video in a scalable way will continue to be an important issue. The video compression technique introduced as part of this work involves transmitting a reference frame periodically. However, given the properties of lecture video, it should be possible to recycle frames and thus avoid completely the need to send reference frames in steady state operation. The obvious benefit of recycling is a reduction in bit rate without compromising subjective quality. To achieve this end, one needs to explore how recycled frames should be selected, develop a strategy to minimize the number of stored reference frames, and consider computationally efficient implementations.

7.2.3 SVC Extension

Scalabe video coding (SVC) has attracted attention with the development of different multimedia applications due to the demand for clients with diverse capabilities in bandwidth, power, complexity and display resolution. It supports multiple spatial, temporal, and SNR scalabilities under the constraints of low complexity and low delay without the need for transcoding.

Although our major goal for the whole framework is not to obtain coding efficiency but to provide a more generic spatio-temporal video interpolation framework, and the mechanism of SVC is fundamentally different from our approach, the proposed work can be integrated into SVC to help improve the spatial scalable coding.

LIST OF REFERENCES

LIST OF REFERENCES

[1] T. M. L. et al., “Survey: Interpolation methods in medical image processing,” *IEEE Trans. Med. Imag.*, vol. 18, pp. 1049–1075, Nov. 1999.

[2] R. G. Keys, “Cubic convolution for digital image processing,” *IEEE Trans. Acous., Speech and signal Process.*, vol. ASSP-29, pp. 1153–1160, Dec. 1981.

[3] M. Unser, “Splines: a perfect fit for signal and image processing,” *IEEE Signal Process. Mag.*, vol. 16, pp. 22–38, Nov. 1999.

[4] T. K. et al., “Picture conversioin apparatus, picture conversion method, learning apparatus and learning method,” *US-patent*, vol. 6, Nov. 2001.

[5] C. B. A. et al., “Optimal image scaling using pixel classification,” *Proc. ICIP 2001*, vol. 3, pp. 864–867, 2001.

[6] X. Li and M. Orchard, “New edge-directed interpolation,” *IEEE Transactions on Image Processing*, vol. 10, pp. 1521–1527, Oct 2001.

[7] T. S. Huang and R. Y. Tsai, “Multi-frame image restoration and registration,” *Advanced Computing Visual Image Processing*, pp. 348–365, 1999.

[8] S. P. Kim, N. K. Bose, and H. M. Valenzuela, “Recursive reconstruction of high resolution image from noisy undersampled multiframes,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, pp. 1013–1027, Jun. 1990.

[9] N. K. Bose, H. C. Kim, and H. M. Valenzuela, “Recursive implementation of total least squares algorithm for image reconstruction from noisy, undersampled multiframes,” in *Proc. IEEE Conf. Acoust., Speech Signal Process*, vol. 5, pp. 269–272, Apr. 1993.

[10] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, “Fast and robust multi-frame super-resolution,” *IEEE Transactions on Image ProcessinG*, vol. 13, no. 10, pp. 1327–1344, Oct. 2004.

[11] B. C. Tom and A. K. Katsaggelos, “Reconstruction of a high-resolution image by simultaneous registration, restoration, and interpolation of low-resolution images,” in *Proceedings of the IEEE International Conference on Image Processing*, pp. 539–542, 1995.

[12] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, “Super-resolution reconstruction of compressed video using transform-domain statistics,” *IEEE Transactions on Image Processing*, vol. 13, no. 1, pp. 33–43, 2004.

[13] C. A. Segall, A. K. Katsaggelos, R. Molina, and J. Mateos, “Bayesian resolution enhancement of compressed video,” *IEEE Trans. Image Processing*, vol. 13, no. 7, pp. 898–911, Jul. 2004.

[14] P. E. Eren, M. I. Sezan, and A. M. Tekalp, “Robust object-based high-resolution image reconstruction from low-resolution video,” *IEEE Transactions on Image Processing*, vol. 6, no. 10, pp. 1446–1451, Oct. 1997.

[15] A. J. Patti and Y. Altunbasak, “Artifact reduction for set theoretic super resolution image reconstruction with edge adaptive constraints and higher-order interpolants,” *IEEE Transactions on Image Processing*, vol. 10, no. 1, pp. 179–186, Jan. 2001.

[16] S. Borman and R. L. Stevenson, “Spatial resolution enhancement of low-resolution image sequences. a comprehensive review with directions for future research,” technical report, Lab. Image and Signal Analysis, University of Notre Dame, 1998.

[17] S. C. P. et al., “Super-resolution image reconstruction: a technical overview,” *IEEE Signal Processing Magzine*, vol. 20, pp. 21–36, May 2003.

[18] R. Molina, A. K. Katsaggelos, and J. Mateos, *Super Resolution of Images and Video*. Morgan & Claypool Publishers, 2007.

[19] E. S. Chaudhuri, *Super-Resolution Imaging*. Norwell,MA:Kluwer, 2001.

[20] M. Ghanbari, “The cross search algorithm for motion estimation,” *IEEE Transaction on Communication*, vol. 38, no. 7, pp. 950–953, 1990.

[21] R. Li, B. Zeng, and M. Liou, “A new three-step search algorithm for block motion estimation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 438–442, Aug. 1994.

[22] M. Bierling, “Displacement estimation by hierarchical block matching,” *Proc. of SPIE Visual Commun. Image Processing (VCIP’88)*, pp. 942–951, 1988.

[23] A. Zaccarin and B. Liu, “Fast algorithms for block motion estimation,” *IEEE Int. Conf. Acoust., Speech, and Signal Processing (ICASSP’92)*, pp. 449–452, 1992.

[24] B. Liu and A. Zaccarin, “New fast algorithms for the estimation of block motion vectors,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, no. 2, pp. 148–157, 1993.

[25] Y. Wang, Y. Wang, and H. Kuroda, “A globally adaptive pixel-decimation algorithm for block-motion estimation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 6, pp. 1006–1011, 2000.

[26] B. K. P. Horn and B. G. Schunck, “Determing optical flow,” *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[27] M. J. Black and P. Anandan, “The robust estimation of multiple motions: Affine and piecewise-smooth flow fields,” technical report, Xerox Systems and Practices Laboratory, 1993. http://www.parc.xerox.com.

[28] R. Szeliski and H. Y. Shum, “Motion estimation with quadtree splines,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 1199–1210, Dec 1996.

[29] J. Monaco, *Generalized Motion Models for Video Applications*. PhD thesis, Georgia Institute of Technology, 1997.

[30] D. H. Frakes, C. P. Conrad, T. M. Healy, J. W. Monaco, M. Fogel, S. Sharma, M. J. T. Smith, and A. P. Yoganathan, “Application of an adaptive control grid interpolation technique to morphological vascular restruction,” *IEEE Transaction on Biomedical Engineering*, vol. 50, Feb 2003.

[31] A. M. Geurtz, J. Biemond, J. N. Driessen, and D. E. Boekee, “A pel-recursive wiener based algorithm for the simultaneous estimation of rotation and translation,” *Proceedings SPIE VCIP*, vol. 1001, pp. 919–924, 1988.

[32] J. Huang, *Motion Estimation and Compensation for Video Image Sequences*. PhD thesis, Georgia Institute of Technology, 1995.

[33] J. Biemond, L. Looijenga, and D. E. Boekee, “A pel-recursive wiener based displacement estimation algorithm,” *Signal Processing*, vol. 13, pp. 399–412, 1987.

[34] H. H. Nagel, “Displacement vectors derive from second-order intensity variations in image sequences,” *CVGIP*, vol. 21, pp. 85–117, Jan 1983.

[35] H. H. Nagel and W. Enkelmann, “An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 565–593, 1986.

[36] E. C. Hildreth, “Computing the velocity field along contours,” *Proceddings of the ACM SIGGRAPH/SIGART Interdisciplinary Workshop on Motion Representation and Perception*, pp. 121–127, 1986.

[37] J. Duncan and T. Chou, “Temporal edges: The detection of motion and the computation of optical flow,” *2nd International Conference aon Computer Vision*, pp. 374–382, 1988.

[38] S. V. Fogel, “The estimation of velocity vector fields from time-varying image sequences,” *CVGIPIU*, vol. 53, pp. 253–287, May 1991.

[39] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *Proceedings of Seventh International Joint Conference on Artificial Intelligence*, (Vancouver, Canada), pp. 674–679, 1981.

[40] D. Fleet and A. Jepson, “Computation of component image velocity from local phase information,” *International Journal of Computer Vision*, vol. 5(1), pp. 77–104, 1990.

[41] M. Black and P. Anandan, “Robust dynamic motion estimation over time,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 292–302, 1991.

[42] H. Nagel, “Constraints for the estimation of idsplacement vector fields form image sequences,” *Proceedings of Eighth International Joint Conference on Artificial Intelligence*, vol. 2, pp. 945–951, 1983.

[43] M. Proesmans, L. V. Gool, E. Pauwels, and A. Oosterlinck, “Determination of optical flow and its discontinuities using non-linear diffusion,” *Proceedings of the Third European Conference - Volume II on Computer Vision (ECCV ’94)*, vol. 2, pp. 295–304, 1994.

[44] S. Uras, F. Girosi, A. Verri, and V. Torre, “Computational approach to motion perception,” *Biological Cybernetics*, vol. 60, pp. 79–87, 1988.

[45] W. V. W.H. Press, S.A. Teukolsky and B. Flannery, *Numerical Recipes in C*. Cambridge University Press, 1992.

[46] J. M. Odobez and P. Bouthemy, "Robust multiresolution estimation of parametric motion models," *Visual Communication and Image Representation*, vol. 4, pp. 348–365, Dec 1995.

[47] C. Lettsome, M. J. T. Smith, and R. Mersereau, "Fixed analysis adaptive synthesis filter banks," *SPIE Security and Defense Symposium*, March 18-20 2007.

[48] C. Lettsome and M. J. T. Smith, "Image interpolation exploiting phase diversity," *IEEE DSP Workshop*, January 2009.

[49] M. J. T. Smith and W. Chung, "Recursive time-varing filter banks for subband image coding," *IEEE Trans. on Signal Processing*, pp. 885–895, July 1995.

[50] I. Sodagar, K. Nayebi, T. Barnwell, and M. J. T. Smith, "Time-varying analysis-synthesis systems based on filter banks and post filtering," *IEEE Trans. on Signal Processing*, pp. 2512–2524, Nov. 1995.

[51] J. Arrowood and M. Smith, "Exact reconstruction analysis/synthesis filter banks with time-varying filters," *Proceedings of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 233–236, April 27-30, 1993.

[52] Y. Chen and M. J. T. Smith, "High quality spatial interpolation of video frames using an adaptive warping method," *IEEE 12th Digital Signal Processing Workshop*, vol. 1, pp. 34–37, Sept 2006.

[53] P. Vandewalle, S. Susstrunk, and M. Vetterli, "A frequency domain approach to registration of aliased images with application to super-resolution," *EURASIP Journal on Applied Signal Processing*, vol. 2006, p. 14, 2006.

[54] Y. Chen and M. J. Smith, "A robust motion estimation method using warping for video frame enlargement," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 841–844, Mar. 2008.

[55] Y. Chen and M. J. Smith, "A concatenated model for video frame interpolation," *13th IEEE Digital Signal Processing Workshop/5th IEEE Signal Processing Education Workshop*, vol. 1 and 2, pp. 565–569, Jan. 2009.

[56] Y. Chen, M. J. T. Smith, and E. Delp, "A low bit-rate video coding approach using modified adaptive warping and long-term spatial memory," *Visual Communications and Image Processing*, vol. 6508, pp. 50803–50803, Jan 2007.

[57] Y. Chen, M. J. Smith, and E. Delp, "An approach to enhanced definition video coding using adaptive warping," *Proceedings of Visual Communications and Image Processing 2009*, Jan. 2009.

[58] M. Annegarn, A. Nillesen, and R. Raven, "Digital signal processing in television receivers," *Philips Technical Review*, vol. 42, no. 6/7, pp. 183–200, April 1986.

[59] G. de Haan, P. Biezen, and O. Ojo, "An evolutionary architecture for motion-compensated 100 hz television," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, no. 3, pp. 207–217, June 1995.

[60] G. de Haan, P. Biezen, H. Huijgen, and O. Ojo, “True motion estimation with 3d recursive search block-matching,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, pp. 368–388, Oct 1993.

[61] O. A. Ojo and G. de Haan, “Robust motion-compensated video upconversion,” *IEEE Transaction on Consumer Electronics*, vol. 43, Nov 1997.

[62] A. Huang and T. Nguyen, “Motion vector processing based on residual energy information for motio compensated frame interpolation,” *Proc. Int. Conf. Image Processing*, vol. 4, pp. 353–356, Sep 2006.

[63] A. Huang and T. Nguyen, “A multistage motion vector processing method for motion-compensated frame interpolation,” *IEEE Transactions on Image Processing*, vol. 17, pp. 694–708, May 2008.

[64] 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.

[65] D. Frakes, L. Dasi, K. Pekkan, H. Kitajima, K. Sundareswaran, A. Yoganathan, and M. J. Smith, “A new method for registration-based medical image interpolation,” *IEEE Transaction on Medical Imaging*, vol. 27, March 2008.

[66] F. Lopes and M. Ghanbari, “Improved motion compensation with overlapped spatial transformation,” *IEEE International Conference on Electronics, Circuits and Systems*, vol. 2, no. 7-10, pp. 457–460, September 1998.

[67] S. Lee and J. Kim, “Fast block motion estimation for overlapped motion compensation using selective pixel matching,” *International Conference on Image Processing*, vol. 11, pp. 80–83, 1999.

[68] H. Watanabe and S. Singhal, “Windowed motion compensation,” *SPIE Visual Communication and Image Processing VCIP*, vol. 1605, pp. 582–589, November 1991.

[69] F. Lopes and M. Ghanbari, “Analysis of spatial transform motion estimation with overlapped compensation and fractional-pixel accuracy,” *IEEE Proceedings Vision, Image and Signal Processing*, vol. 146, no. 6, pp. 339–344, December 1999.

[70] A.-M. Huang and T. Nguyen, “Correlation-based motion vector processing for motion compensated frame interpolation,” *15th IEEE International Conference on Image Processing*, pp. 1244–1247, Oct 2008.

[71] Y. Chen, M. J. Smith, and E. Delp, “Adaptive motion estimation using warping for video frame rate up-conversion,” *Proceedings of Visual Information Processing and Communication 2010*, Jan. 2010.

VITA

VITA

Ying Chen Lou was born in Shanghai, P.R. China, in 1982. She earned her Bachelor of Science Degree in Electrical Engineering with high honor from Shanghai Jiao Tong University in 2004. In fall 2003, she was an exchange student at Purdue University where she decided to start her graduate school later on.

Ying began her graduate study at Purdue University in 2004. After a year, she joined the Video and Image Processing Laboratory (VIPER) as a research assistant under the supervision of Professors Mark J.T. Smith and Edward J. Delp.

She augmented her academic background with summer internships at Apple Inc. in 2006, Texas Instruments in 2007, 2008, and Qualcomm Inc. in 2010. During internships, she worked for several projects such as face detection, rate control, memory compression, and tracking projects.

She has completed her PhD dissertation in 2010. Her research interests include video and image compression (JPEG/JPEG2000/MPEG-2/H.263/MPEG-4/H.264/AVC), image and video post-processing, tracking, and camera (ISP) related work such as denoising, demosaicing, autofocus.