Single and Multi-view Video Super-resolution

SINGLE AND MULTI-VIEW VIDEO SUPER-RESOLUTION

BY

SEYEDREZA NAJAFI, M.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

Master of Applied Science

Master of Applied Science (2012) McMaster University

(Electrical & Computer Engineering) Hamilton, Ontario, Canada

TITLE:	Single and Multi-view Video Super-resolution
AUTHOR:	Seyedreza Najafi M.Sc., (Electrical Engineering) Tehran Polytechnic, Tehran, Iran
SUPERVISOR:	Dr. Shahram Shirani
NUMBER OF PAGES:	ix, 87

Abstract

Video super-resolution for dual-mode cameras in single-view and mono-view scenarios is studied in this thesis. Dual-mode cameras are capable of generating high-resolution still images while shooting video sequences at low-resolution. High-resolution still images are used to form a regularization function for solving the inverse problem of super-resolution. Exploiting proposed regularization function in this thesis obviates the need for classic regularization function. Experimental results show that using proposed regularization function instead of classic regularization functions for super-resolution of single-view video leads to improved results. In this thesis, super-resolution problem is divided into low-resolution frame fusion and de-blurring. A frame fusion scheme for multi-view video is proposed and performance improvement when exploiting multi-view sequence instead of single-view for frame fusion is studied. Experimental results show that information taken by a set of cameras instead of a single camera can improve super-resolution process, especially when video contains fast motions. As a side work, we applied our low-resolution multi-view frame fusion algorithm to 3D frame-compatible format resolution enhancement. Multi-view video super-resolution using high-resolution still images is performed at the decoder to prevent increasing computation complexity of the encoder. Experimental results show that this method delivers comparable compression efficiency for lower bit-rates.

Acknowledgements

I would like to express my gratitude to my supervisor, Dr. Shahram Shirani, whose expertise, and understanding added considerably to my graduate experience. I appreciate his kindness, and patience as my supervisor, and his assistance in writing papers and this thesis.

Contents

List of Figures

Chapter 1

Introduction and Problem Statement

A video camera is required to deliver a video sequence at desired frame-rate and spatial resolution. Fulfilling this demand is a challenge for some applications due to physical limitations of imaging systems. Obviously, high-resolution, high frame-rate video of a scene is desirable because it contains more recognizable details, and is more pleasant. A trade off exists between frame-rate and resolution, where improving both at the same time is either not possible or leads to an expensive or heavy imaging device, which is not practical for many applications. As an instance, consider a consumer video recorder that uses CMOS image sensors. To increase resolution, one approach is to increase pixel density of the image sensor. This may decreases the Signal to Noise Ratio (SNR) of the sensor output, when the size of the sensor remains the same and thus the area of each pixel on the chip decreases. As a result, the image sensor size should be increased, that increases the size of optics used in the device, and also increases the recording time for each frame because capacitance of CMOS

sensors increases with their size [Park *et al.* (2003)]. Increasing recording time for each frame is equivalent to a lower frame-rate. Although this trade off may be solved by employing cutting edge of semiconductor technology, the associated cost usually renders that unsuitable for consumer applications. As a result, most of digital video cameras produce low-resolution videos at standard frame-rate. What if we need high-resolution video sequence in an application and high-quality imaging devices are not practical? Signal processing approaches may be employed to deliver desired video sequence from a non-ideal imaging system. These approaches may be adopted for an imaging application to keep the cost of the system acceptable, and render low-resolution imaging devices an option for more demanding applications [Park *et al.* (2003)].

The problem of enhancing each frame of a low-resolution video sequence by exploiting information of many adjacent frames is an interesting and well-researched subject in the area of signal processing, which is called "multi-frame super-resolution". In applications such as medical imaging or remote sensing, a high frame-rate sequence of the object may not be generally a must. While desirable high-resolution images are not achievable using the available device, a sequence of frames may be used for generating one high-resolution image. In fact, multi-frame super-resolution algorithms were proposed basically for producing one single high-resolution image of the scene by taking more than one frame and then combining their information. Therefore, super-resolution techniques can be applied to a wide variety of applications and are not limited to the video enhancement, and have been shown to lead to promising results [Park *et al.* (2003)].

1.1 Overview of the Thesis

In the second chapter, we review the frame formation process in a camera and the related works in the area of mono sequence super-resolution is studied.

Extensive research has been done on super-resolution of a single low-resolution sequence. Less work has been done to exploit more than one sequence for super-resolution. In this thesis we examine different scenarios for video super-resolution. First, we assume that the camera is capable of producing a sequence of high-resolution images at a large time interval (*e.g.* 10 frame of video sequence) while shooting low-resolution video. This camera may be called a dual-mode camera, which is capable of performing both tasks simultaneously [X. Wu (2010)]. Images of this sequence are called auxiliary still images in this thesis. Using such a sequence of images has been studied before [B. Song (2011)]. In the third chapter, we propose a super-resolution algorithm to exploit these images and form a regularization function for solving the inverse problem of the super-resolution process, which is new in this area [S. Najafi (2012)].

Next, we examine how using multiple low-resolution cameras, instead of one, may enhance the super-resolution process in the chapter four. The sequence resulted from recording the scene by multiple cameras is a multi-view video. Efficient fusion of multi-view video frames is studied for enhancing super-resolution process. As a side work, the basics of the algorithm proposed for exploiting multi-view video in the super-resolution process are used for stereo video de-interlacing.

Finally in chapter four, we extend the idea of using auxiliary still images from single-view to multi-view video and study its performance. This thesis is concluded in the fifth chapter.

Chapter 2

Multi-frame Super-resolution Background

In this chapter, an overview of LR frame formation in digital cameras, and principals of multi-frame super-resolution for single sequences is presented in this chapter. Sources of degradation, and mathematical model for Low-Resolution (LR) and High-Resolution (HR) frame formation is described briefly.

Multi-frame super-resolution problem can be divided into frame fusion and restoration tasks. Figure 2.1 illustrates the process ... . Frame fusion consists of registration of adjacent frames and combining them to reconstruct a primary up-sampled frame, which is studied in the section 2.3. Primarily up-sampled frame should be de-noised and de-blurred in order to achieve the final super-resolved frame. This problem is studied in section 2.2.1.

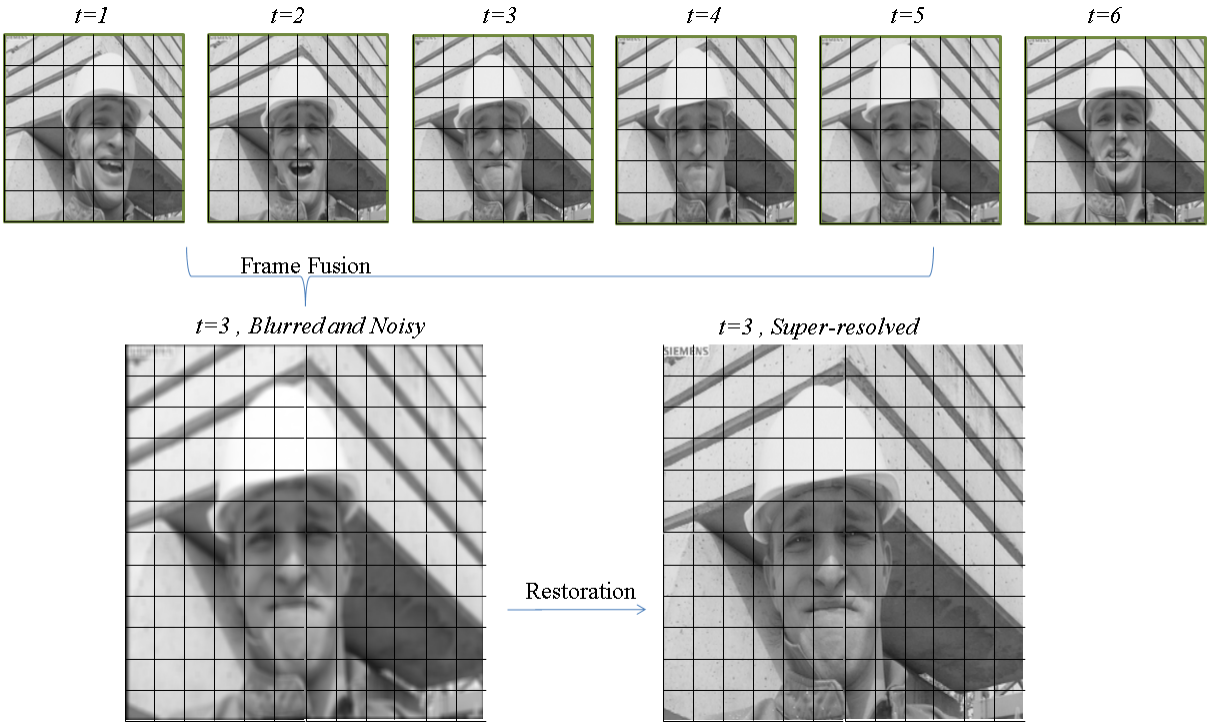

Figure 2.1: Multi-frame super-resolution process.

2.1 Low-resolution Frame Formation

A digital video sequence is in fact the result of 3-dimensional time-space sampling of the scene. Each frame is a 2D representation of the scene, spatially sampled at a time instance. Multiple frames along time axis add the third dimension to the sequence, representing temporal sampling of the scene. Image sensor plays the important role of spatial sampling of the scene. Image sensor is a 2D array of CCD or CMOS light sensor units. Each unit converts the photons detected on its surface into electric signals. The number of light sensor units on the camera image sensor determines the maximum achievable spatial resolution of the camera. Camera electronic or mechanical shutter speed determines the frame-rate and exposure time. These characteristics play an important role in the quality of recorded frames. In addition to these two factors, many other factors such as size, pixel density, and dynamic range of image sensor (which is the ratio of the largest and smallest luminance detectable by the sensor), affect the final quality of recorded frame sequence.

The signal recorded by each light sensor unit is directly related to the number of photons detected on its surface. This number is related to the average light intensity in the corresponding area, the exposure time, and the area of each light sensor. The number of photons received from each point of the scene determines the accuracy of its intensity estimation in presence of different sources of noise. The exposure time must be sufficient to let a minimum number of photons arrive at the image sensor surface to produce an accurate measurement with a reasonable SNR; as a result, the maximum achievable frame-rate, which is inversely related to the exposure time, is limited. This maximum could increase if the light sensor area is increased and/or scene illumination is enhanced, but these are not always practical. Increasing the sensor

size means that larger optical devices are required, moreover a larger and heavier camera is not capable of near-field focus [Pelletier *et al.* (2005)]. Scene illumination adjustment is not achievable in every situation. As a trade off, in multi-resolution image sensors, spatial resolution may be sacrificed to achieve higher frame-rates. In this approach, a fraction of maximum spatial resolution of image sensor is achieved at a higher frame-rate by accumulating photons detected over a larger area of the image sensor for each pixel. This method is called "Binning" and is popular for both CCD and CMOS sensors [Huang *et al.* (2011)]. In this way, image sensor noise is reduced by accumulating weak signals of adjacent pixels on-chip. Usually groups of adjacent 2×2 or 4×4 or even 16×16 pixels merge together to increase output SNR. For instance, the SNR of the CCD image sensor output increases either by the square root, or linearly with the number of pixels binned, depending on which type of noise is dominant in the image sensor [Zhou *et al.* (1997)]. Although it seems to be possible to perform averaging operation off-chip, on-chip binning is dramatically more effective because sources of noise is suppressed directly on the image sensor chip before read-out in the case of on-chip binning. Therefore, image sensor has two or more grids to define area of each pixel. HR grid associates measurement of one light sensor to each output pixel, and LR grid associates many adjacent light sensor to each pixel (See Figure 2.2). In the LR grid mode, accumulation of measurements by many adjacent light sensors lead to a reasonable record of scene intensity in terms of signal to noise ratio but with a lower resolution compared to HR grid mode as is shown in Figure 2.3 . Many consumer digital camcorders employ the LR grid mode while shooting video at high frame-rates as exposure time is limited. In contrary, image sensor's HR grid is used for taking still images, where exposure time is more flexible [X. Wu (2010)].

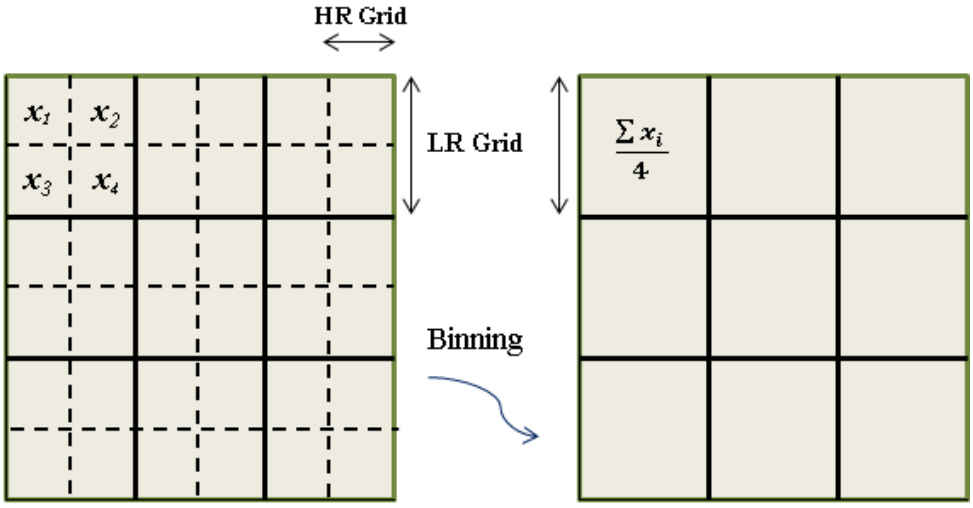

Figure 2.2: Pixel binning, LR and HR grid of image sensor.

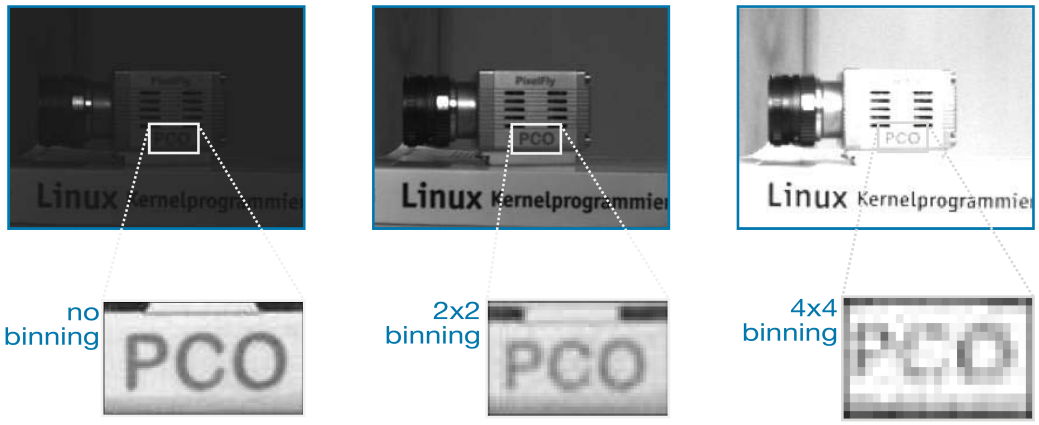

Figure 2.3: Effect of pixel binning, [PCO-TECH (2012)]. From left to right: No binning, 2×2 pixel binning, 4×4 pixel binning. Note: Contrast of magnified portions is enhanced for better illustration.

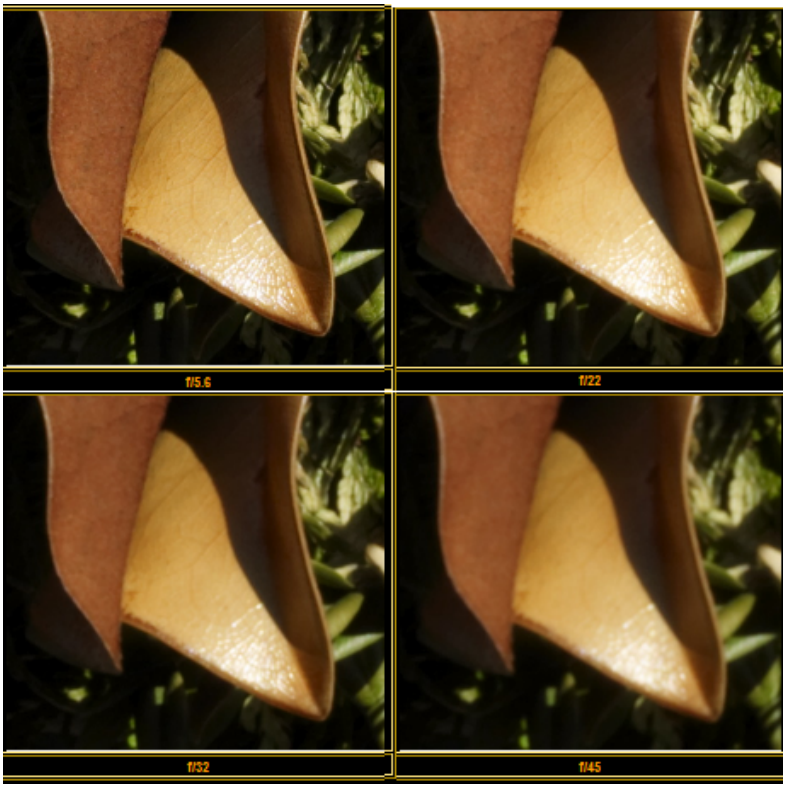

Figure 2.4: Effect of lens and optic devices for different aperture sizes [LuminousLandscape (2012)]. Top-left: f/5.6 aperture size, Top-right: f/22 aperture size, down-left: f/32 aperture size, down-right:f/45 aperture size. The smaller the aperture size, the more visible lens diffraction effect.

Since each pixel value produced by LR grid includes accumulation of signals recorded by adjacent light sensors, resulting frame apparently suffers from blurring followed by down-sampled when compared to the frame that could have formed on the HR grid.

Pixel binning is only one blurring source influencing the output frame. Out-of-focus blur, and diffraction related to the optic parts of camera such as the lens and aperture (as is shown in Figure 2.4) are other sources of blurring [Park *et al.* (2003)].

Point Spread Function (PSF) of imaging system includes all blurring and distortions affecting each point of the scene during the process of frame formation. PSF of an imaging system describes the response of imaging system to a point light source.

The concept of PSF may be interpreted as the impulse response of the imaging system and determines important characteristics of the system. PSF is theoretically described as a continuous finite impulse response, which convolves with the signal of light intensities of the scene to produce the resulting frame. If the PSF remains the same throughout the domain of the formed frame, the convolution of real scene by it may be interpreted as a linear filtering process. The aforementioned filtered frame undergoes down-sampling due to finiteness of density of light sensors on the image sensor. Although filtering by PSF function and down-sampling process happen simultaneously in digital cameras, these operation may be modeled separately. PSF operates on the scene signal as the input that is ideally continuous. Assume $\boldsymbol{x}(m,n)$ as a continuous 2D signal representing scene luminance at each (m,n) location. If we show PSF function of a digital imaging system as $h(m,n)$, then the frame formation equation on this system will be as follows:

$$z(m,n) = h(m,n) * x(m,n)$$
$$\boldsymbol{y}(i,j) = z(iL, jL) + \boldsymbol{n}(i,j)$$

where $z(m,n)$ is the continuous blurred frame before sampling and reading as digital data by the image sensor read-out circuitry. $\boldsymbol{y}(i,j)$ is the digital frame after contaminating by different sources of noise $\boldsymbol{n}(i,j)$, and being sampled along the horizontal and vertical axes on the image sensor grid. Different sources of image sensor noise including "Shot Noise", "Reset Noise" (for CMOS image sensors), and "Readout" noise which also includes quantization noise contaminate the formed frame [Liu and El Gamal (2003)]. L represents the width and hight of each pixel on the image sensor. This frame formation model is not useful since it relates the observed signal to the

continuous signal representing the scene. As we do not want to reconstruct the ideal continuous signal of the scene, we should find an equation relating observed LR frame formed on the LR grid of imaging device to the frame that could have been formed on the HR grid of that device. The PSF relating these two digital frames is a discrete function that could be represented by a matrix, and can relate the LR and HR frames by a discrete convolution operation:

$$\boldsymbol{z} = \boldsymbol{h} * \boldsymbol{x}$$
$$\boldsymbol{y} = D\{\boldsymbol{z}\} + \boldsymbol{n}$$

where D is down-sampling operator. Consider $\boldsymbol{y}$ is formed on $M \times N$ LR grid of image sensor, and $\boldsymbol{x}$ is the image could have formed on HR grid of size $rM \times rN$. By re-arranging $\boldsymbol{y}$ and $\boldsymbol{x}$ in lexicographic order, and reshaping $\boldsymbol{h}$ from a blurring window to a circulant blurring matrix $\boldsymbol{H}$ of size $r^2MN \times r^2MN$ we can re-write the above equation as a linear relationship between $\boldsymbol{x}$ and $\boldsymbol{y}$:

$$\boldsymbol{y} = \boldsymbol{DHx} + \boldsymbol{n} \tag{2.1}$$

where $\boldsymbol{x}$ is of size $r^2MN \times 1$ and $\boldsymbol{y}$ is of size $MN \times 1$. $\boldsymbol{D}$ is a short matrix of size $MN \times r^2MN$ playing the role of down-sampling. $\boldsymbol{n}$ is assumed to be a zero-mean noise with variance equal to σ^2.

Multi-frame super-resolution problem we address in this thesis deals with recovering $\boldsymbol{x}$ that has been degraded through this model. In the next section, recent approaches to super-resolution frame reconstruction are presented.

2.2 Super-resolution Reconstruction

Consider $\boldsymbol{y}$ is the current LR frame we want to reconstruct. Focusing on equation 2.1, the reconstruction problem can be considered as up-sampling, de-blurring and de-noising of $\boldsymbol{y}$ to find an estimate to $\boldsymbol{x}$, which is the original HR frame. Recovering $\boldsymbol{x}$ from this equation directly is not possible because of the following argument. Degradation operator $\boldsymbol{DH}$ is a rank deficient matrix for two reasons:

1-short form of $\boldsymbol{D}$,

2-rank-deficiency of $\boldsymbol{H}$.

$\boldsymbol{D}$ renders the problem under-determined. In other words, the goal is to find r^2NM unknowns, when we have NM equations. $\boldsymbol{H}$ is a circulant matrix, because PSF is considered to be space-invariant. Therefore, its eigenvalues are Discrete Fourier Transform (DFT) coefficients of its first row [Davis (1979)]. Moreover, PSF of the image sensor can be represented as an averaging window which weights the surrounding pixels of each pixel, and averages them to form the LR grid pixel value. Therefore, PSF is low-pass in nature. DFT coefficients of its impulse response is zero for larger indexed coefficients (*i.e.* high-frequency coefficients) which means some eigenvalues of $\boldsymbol{H}$ are zero, and it is rank deficient. As a result, the inverse problem is ill-posed therefore can not be solved using only information given by $\boldsymbol{y}$ and other information should be employed. The other issue is the effect of the noise, $\boldsymbol{n}$. How the recovered $\boldsymbol{x}$ may be kept away from the noise contaminating $\boldsymbol{y}$? Solution to all of these problems lies in the concept of regularization function which is based on the prior information about desired frame. Before trying to use priors for recovering the current frame, $\boldsymbol{x}$, all the relevant information from observations has to be included in the formulations. Adjacent frames may carry precious information for recovering samples removed from

the HR grid of current frame. For this aim, we assume current frame to be the original representation of the scene, and pixels of the adjacent frames to be translated and deformed version of the current frame. Therefore, motion of those pixels have to be determined and taken into account. To use adjacent frames' pixels, they should be accurately registered on the HR grid of current frame. Assume registration from adjacent frame's HR grid to the current frame's HR grid is possible, and can be done by a linear operation and $\boldsymbol{F}_t$ is the warping matrix relates the current frame to t^{th} adjacent frame, therefore,

$$\boldsymbol{y}_t = \boldsymbol{DHF}_t\boldsymbol{x} + \boldsymbol{n}$$

where $\boldsymbol{y}_t$ is t^{th} adjacent frame to current frame. Let's for generalization assume that current frame is one of $\boldsymbol{y}_t$s and its formation is associated with a warping matrix (which is basically represents no motion). Consider we want to include T frames in the super-resolution process of the current frame, and $\boldsymbol{F}_t$ is the warping matrix which maps pixel locations on the HR grid of current frame to the proper location on the HR grid of each adjacent frame. Therefore, we will have T equations for the unknown $\boldsymbol{x}$:

$$\boldsymbol{y}_t = \boldsymbol{DHF}_t\boldsymbol{x} + \boldsymbol{n} \quad 1 < t < T$$

Considering only the information of adjacent frames, we may take the Least Squares (LS) approach to restore $\boldsymbol{x}$. At this point, by assuming equal noise variance for all adjacent frames, LS answer to the above equation set is as follows:

$$\begin{aligned} e_{LS}(\boldsymbol{x}) &= \sum_{t=1}^{T} \|\boldsymbol{DF}_t\boldsymbol{Hx} - \boldsymbol{y}_t\|_2^2 \\ \hat{\boldsymbol{x}} &= \min_x\{e_{LS}(\boldsymbol{x})\} \end{aligned} \tag{2.2}$$

This solution minimizes the square error related to each equation, and is the same when we take the Maximum Likelihood (ML) approach to this problem. In terms of ML criterion, $\boldsymbol{P}(\boldsymbol{y}|\boldsymbol{x})$ should be maximized, and it results in the same formulation assuming white Gaussianity for the noise. As stated before, because of rank deficiency of $\boldsymbol{H}$ and $\boldsymbol{D}$, solution to (2.2) is not stable or unique. To show the effect of rank deficiency of $\boldsymbol{H}$ on the solution, assume $\boldsymbol{x}_{opt}$ is the un-degraded HR frame which is naturally an answer to (2.2):

$$e(\boldsymbol{x}_{opt}) \leq e(\boldsymbol{x}), \forall \boldsymbol{x}$$

In addition, assume $\boldsymbol{x}_{hf}$ is a high frequency noise existing in the null-space of $\boldsymbol{H}$, then

$$\boldsymbol{H}\boldsymbol{x}_{hf} = \boldsymbol{0}$$

$$e(\boldsymbol{x}_{opt} + \boldsymbol{x}_{hf}) = e(\boldsymbol{x}_{opt})$$

and $\boldsymbol{x}_{opt} + \boldsymbol{x}_{hf}$ is another answer to (2.2):

$$e(\boldsymbol{x}_{opt} + \boldsymbol{x}_{hf}) \leq e(\boldsymbol{x}), \forall \boldsymbol{x}.$$

Therefore, any combination of un-degraded signal $\boldsymbol{x}$ as the true answer for (2.2), and a high-frequency noise from null-space of $\boldsymbol{H}$ can be another undesirable answer of (2.2). This problem does not have a unique solution and for example; the solution changes by a minor alteration of the observations due to noise. A constraint should be added to limit the restored signal to a desirable range, and minimizes its high-frequency energy content. This knowledge is added to the problem as a "regularization function" for stabilizing the solution, or as an "image prior" in a stochastic approach. Although

this primary least squares formulation we discussed here cannot lead us to a stable answer, it is the core of final formulation and has the very important role of frame fusion in super-resolution process. This cost function defines how adjacent frames should be mixed with each other and contribute to super-resolution of current frame. After studying and reviewing regularization functions for this inverse problem, we will take a deeper look at this primary answer in section 2.3, and discuss how it impacts the final reconstructed frame.

2.2.1 Regularized Reconstruction Approach

Basically, a regularization function is built based on general ideas about how a natural image should look like. Regularization function, $\Psi(\boldsymbol{x})$, returns large values for irrelevant inputs and is added to former formulation (2.2) using γ as a Lagrangian multiplier:

$$\begin{gathered} e(x) = \sum_{t=1}^{T} \|\boldsymbol{D}\boldsymbol{F}_t\boldsymbol{H}\boldsymbol{x} - \boldsymbol{y}_t\|_2^2 + \gamma\Psi(\boldsymbol{x}) \\ \hat{\boldsymbol{x}} = \min_x\{e(\boldsymbol{x})\} \end{gathered} \tag{2.3}$$

This formulation may be achieved by easily applying MAP estimation to the problem and using the following function as the image prior:

$$P(\boldsymbol{x}) = exp(-\Psi(\boldsymbol{x}))$$

As can be seen, the regularization function appears in an exponential function's argument to play the role of Probability Distribution Function (PDF) of image. Derivation of (2.3)using MAP by maximizing $P(\boldsymbol{x}|\boldsymbol{y})$ is straight-forward.

The classic priors used as regularization function are based on one assumption.

The assumption is that images consist of smooth regions and total energy of high-frequencies are limited. Well-known Tikhonov regularization function and the Total Variation (TV) are also based on this assumption. Tikhonov regularization function is simply Laplacian of the image and can be represented by a linear operator:

$$\begin{aligned} \Psi(\boldsymbol{x}) &= \|\boldsymbol{L}\boldsymbol{x}\|_2^2 \\ \boldsymbol{L} &= \frac{1}{8}\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \end{aligned} \tag{2.4}$$

Laplacian operator is discrete approximation of its continues form. Continues Laplacian operator is defined as:

$$\nabla^2 = (\frac{\partial^2}{\partial u^2} + \frac{\partial^2}{\partial v^2}) \tag{2.5}$$

For image processing problem, Laplacian of Gaussian is used consisting of first filtering image by a Gaussian filter, and then taking second derivatives. Gaussian filter is applied to reduce effect of high-frequency noises. Therefore, continuous Laplacian of Gaussian operator is:

$$\boldsymbol{L} = \nabla^2(G) = -\frac{1}{\pi\sigma^4}(1 - \frac{u^2+v^2}{2\sigma^2})\exp(-\frac{u^2+v^2}{2\sigma^2}) \tag{2.6}$$

Other regularization functions have been used taking derivative of the image as a measure of smoothness. For example the following regularization function may also be used for this aim:

$$\Psi(\boldsymbol{x}) = \|\boldsymbol{D}_h\boldsymbol{x}\|_2^2 + \|\boldsymbol{D}_v\boldsymbol{x}\|_2^2 \tag{2.7}$$

where $\boldsymbol{D}_h$ and $\boldsymbol{D}_v$ are operator matrices taking derivative of image in horizontal and

vertical direction. The type of derivative used in equation (2.7) is first order and directional compared to Tikhonov which is the summation of second order derivatives. Since an image show significantly different values across its edges, this regions are considered as not-desirable outcome, and are penalized using the above regularization functions. Therefore, edges are not recovered properly in the restored image. Specifically penalizing using L2-norm, renders the situation worse. Total Variation (TV) regularization function is well-known for maintaining smoothness while preserving edges of the image. TV is defined as the L1-norm of magnitude of gradient of image [Chan *et al.* (2001)]:

$$\Psi_{TV}(\boldsymbol{x}) = \|\nabla \boldsymbol{x}\|_1 \tag{2.8}$$

Bilateral Total Variation is a modified version of TV, and shown to be very efficient for preserving edges of the image [Farsiu *et al.* (2004)] by employing several scales of derivatives:

$$\Psi_{BTV}(\boldsymbol{x}) = \sum_{i=-P}^{P} \sum_{\substack{j=0 \\ i+j>0}}^{P} \alpha^{(i+j)} \left\| \boldsymbol{x} - \boldsymbol{S}_x^i \boldsymbol{S}_y^j \boldsymbol{x} \right\|_1^1 \tag{2.9}$$

where α is weighting factor used to discriminate between different scales of derivatives, and $\mathbf{S}_x^m$ and $\mathbf{S}_y^m$ are two operators shift the image by m pixel horizontally and vertically, respectively.

Approximation theory may also be employed for devising effective regularization functions. Sparsity among wavelet coefficients which exists in all natural images can be simply put in a closed mathematical expression and used as a regularization function [Donoho and Johnstone (1994)]. This idea comes from observations telling us wavelet transform coefficients are mostly zero or close to zero and tend to be sparse.

To penalize any non-sparse answer, regularization function is formed by taking Lp-norm of the wavelet transform of image,

$$\Psi_W(\boldsymbol{x}) = \| \boldsymbol{T}_W \boldsymbol{x} \|_p^p, p \leq 1 \tag{2.10}$$

where $\boldsymbol{T}_W$ is generally the wavelet operator matrix that may produce coefficients of any high-frequency sub-band needed, when is multiplied by the signal. Although it looks different from previous forms which take derivative of image, the latter regularization function is also based on general assumptions about natural images.

2.3 Conventional Frame Fusion Method

Frame fusion is adding information of adjacent frames in the super-resolution process, and is implicitly embedded in the solution for multi-frame super-resolution presented in equation (2.3). Information of all LR frames contributing to super-resolution of current frame are being added together by proper warping matrices, $\boldsymbol{F}_t$.

To see how frame fusion happens by minimizing (2.3), we separately minimize the first term of the cost function in (2.3), which is equal to

$$\sum_{t=1}^{T} \| \boldsymbol{D}\boldsymbol{F}_t\boldsymbol{H}\boldsymbol{x} - \boldsymbol{y}_t \|_2^2,$$

and is responsible for frame fusion. Assuming $\boldsymbol{z} = \boldsymbol{H}\boldsymbol{x}$ and then minimizing the following expression for $\boldsymbol{z}$:

$$\sum_{t=1}^{T} \| \boldsymbol{D}\boldsymbol{F}_t\boldsymbol{z} - \boldsymbol{y}_t \|_2^2$$

In this way, we are looking at the problem as two separate problems. First, adjacent

frames registration and up-sampling to achieve a blurred noisy up-sampled frame, $\hat{\boldsymbol{z}}$ as follows:

$$\begin{aligned} e_f(\boldsymbol{z}) &= \sum_{t=1}^{T} \|\boldsymbol{D}\boldsymbol{F}_t\boldsymbol{z} - \boldsymbol{y}_t\|_2^2 \\ \hat{\boldsymbol{z}} &= \min_x\{e_f(\boldsymbol{z})\} \end{aligned} \tag{2.11}$$

where $e_f(\boldsymbol{z})$ is the fusion cost function. Second, de-blurring and de-noising of the up-sampled frame are performed by minimizing the following cost function:

$$\begin{aligned} e_r(\boldsymbol{x}) &= \|\boldsymbol{H}\boldsymbol{x} - \hat{\boldsymbol{z}}\|_2^2 + \gamma\Psi(\boldsymbol{x}) \\ \hat{\boldsymbol{x}} &= \min_x\{e_r(\boldsymbol{x})\} \end{aligned} \tag{2.12}$$

where $e_r(\boldsymbol{z})$ is the restoration cost function. It is worth mentioning, both problems are actually solved simultaneously by minimizing (2.3). We only discuss the first minimization problem in this section as it is related to frame fusion and up-sampling.

Solving (2.11) for $\boldsymbol{z}$, is as follows:

$$\frac{de_f(\boldsymbol{z})}{d\boldsymbol{z}} = \frac{d}{d\boldsymbol{z}}\sum_{t=1}^{T}(\boldsymbol{D}\boldsymbol{F}_i\boldsymbol{z} - \boldsymbol{y}_t)^T(\boldsymbol{D}\boldsymbol{F}_i\boldsymbol{z} - \boldsymbol{y}_t) = 0.$$

Taking the derivative, we continue as follows:

$$\sum_{t=1}^{T} 2{\boldsymbol{F}_t}^T\boldsymbol{D}^T(\boldsymbol{D}\boldsymbol{F}_t\boldsymbol{z} - \boldsymbol{y}_t) = 0$$

$$\sum_{t=1}^{T} \boldsymbol{F}_t^T\boldsymbol{D}^T\boldsymbol{D}\boldsymbol{F}_t\boldsymbol{z} = \sum_{t=1}^{T} \boldsymbol{F}_t^T\boldsymbol{D}^T\boldsymbol{y}_t \tag{2.13}$$

This equation shows how adjacent frames are fused for up-sampling the current frame. Assume, HR grid is a $rM \times rN$ matrix of pixels, and LR grid is a $M \times N$ matrix and $r = 2$. Operator $\boldsymbol{D}$ takes pixels located on the odd indexed rows and columns

and put them on the LR grid in the same order. Operator $\boldsymbol{D}^T$ performs the inverse task. It makes a $rM \times rN$ all-zero matrix, takes the pixels of LR frame, put them on the odd indexed rows and columns of the all-zero matrix. Therefore, $\boldsymbol{D}^T\boldsymbol{y}_t$ is simply zero-inserted up-sampled version of $\boldsymbol{y}_t$. This task is performed for each LR frame and its samples placed on the HR grid. $\boldsymbol{F}_t^T$ performs the inverse shift to return the pixels of $\boldsymbol{y}_t$ to their proper position in the current frame, and compensates the relative motion between two frames. The essential requirement for adjacent frames to contribute in up-sampling of current frame constructively, lies in the form of relative motion. If motion helps to shift and put pixels of adjacent frames in a manner that no pixel position of HR grid remains zero in the result of frame fusion, $\sum_{t=1}^{T} \boldsymbol{F}_t^T \boldsymbol{D}^T \boldsymbol{y}_i$, then it is effective. As a result, if less than $r^2 - 1$ adjacent frame are fused (*i.e.* $T < r^2$), some pixel locations on the HR grid remain zero. The most important issue, is capability of algorithm to perform sub-pixel motion estimation and form $\boldsymbol{F}_t$ matrices, and existence of sub-pixel motion between current frame and adjacent frames. Exploiting more than $r^2 - 1$ adjacent frames increases the chance for each pixel location on the HR grid to find a sample in adjacent frames. Possibly, more than one sample could be found for each pixel location on the HR grid. In those cases, averaging is performed on samples of adjacent frames assigned to that location to determine its pixel value. Averaging is the case, when L2-norm is used in (2.2). It is proved in [Farsiu *et al.* (2004)], that in case of using L1-norm as distance measurement between restored frame and degradation process, median is taken between samples of different LR frame associated with a single location to set the pixel value. Using L1-norm has the advantage of outlier rejection, while L2-norm leads to averaging and is effective for mitigating degradation effect due to Gaussian noise.

Although we looked at up-sampling and de-blurring tasks separately in this chapter for better explanation, in many multi-frame super-resolution approaches these tasks are performed simultaneously by minimizing one cost function [Farsiu *et al.* (2004)]. On the other hand, in some methods which adjacent frame fusion is complicated fusion step may be performed separately through a different cost functions for practical implementation issues [Protter *et al.* (2009)]. It should be mentioned that separation of these tasks is sub-optimal but useful in terms of practical implementation.

Conventional motion compensation for frame fusion assigns one motion vector per adjacent frames to each pixel in current frame. It assumes that each point of the scene that is represented by one pixel in each of the adjacent frames, takes integer multiples of HR grid pixel size and moves to another pixel of current frame's HR grid. Then, based on the type of the norm which is used in the fusion term of solution, we take median or mean of pixels associated by motion vectors in adjacent frames. Finding warping matrices on the HR grid requires sub-pixel motion estimation using LR frames, and is one of the challenges that multi-frame super-resolution faces. How could an algorithm estimate sub-pixel motion estimation accurately? The answer is that conventional multi-frame super-resolution algorithms capability is limited to scenes with simple motions, such as global transition. Accurate motion estimation enhances the fusion of LR frame, and non-accurate motion fields leads to sever degradation and artifacts in the super-resolved frame. As a result, recently many efforts have been made to mitigate the need for explicit sub-pixel motion estimation [Takeda *et al.* (2009), Protter *et al.* (2009)]. Frame fusion method presented in [Protter *et al.* (2009)] will be the heart of algorithm presented in next chapter. In the next chapter

we introduce an implicit motion compensation method for frame fusion, which does not assume this simple motion model for the motion of the points of the scene, and assigns more than one motion vector per adjacent frame to each of the pixels in the current frame.

Chapter 3

Video Super-Resolution using Auxiliary High Resolution Still Images

3.1 Introduction

Limitations related to the factors reviewed in the previous chapter in addition to limitations on processing power of camera video encoders in most consumer applications, result in a low resolution (LR) video sequence. Delivered low-resolution frames have gone through different degradation processes including blurring, down-sampling, and contamination by noise. Super-resolution, the problem of reconstructing high resolution (HR) frame from a LR video sequence, is an ill-posed inverse problem and requires extra knowledge about the unknown to be solved. As studied in details in previous chapter, this knowledge is added to the problem as a "regularization function" for stabilizing the solution, or as an "image prior" in a stochastic approach.

Regularization function plays an important role in solving the inverse problem at hand and the recovered signal inherits its characteristics from what was imposed on it by the regularization function.

Although conventional image priors have evolved significantly, the quality of results is still poor since one general characteristic put in terms of simple mathematical expressions cannot represent divers characteristics of images. As a solution, example-based priors have emerged, tuned for a specific family of images. Example-based priors are specified using many examples of HR images with similar content to the image being super-resolved [M. Elad (2007)]. A similar technique is applicable in the area of video super-resolution. It consists of using several HR still images taken from the same scene to enhance the process of video super-resolution. Since the content of video changes over time and HR still images should have a minimum correlation with the video, images should be taken at a regular frequency along the video frames. For example, they may be taken with a rate far smaller than frame rate of the video providing useful information to enhance video super-resolution process.

It is worth mentioning that "video" and "auxiliary HR still image" are two terms used in this chapter which refer to for the LR frame sequence we want to super-resolve and the extra information we want to exploit for the super-resolution, respectively. The sequence of HR still images taken for enhancing super-resolution may be considered as a low frame-rate HR sequence. Super-resolution process should be performed before final coding and recording video on the memory of camera. Nevertheless, we use the term "video" for the LR sequence we want to super-resolve, and "still image" for the extra HR shots of the scene we use for enhancing super-resolution process.

Exploiting information of HR still images, which is practical in a variety of scenarios, have been proposed for video super-resolution enhancement. Using recently developed CMOS image sensors, it is possible for the image sensor chip to generate high-resolution images at a lower rate, while recording high-frame rate low-resolution frame sequence [Pelletier *et al.* (2005)]. This combination may be exploited inside the camera for the LR frame-sequence super-resolution (before compression and storage). In another scenario, we may perform video super-resolution using HR still images using a software on the computer. In that case, compression effect should be considered in the process of super-resolution.

Super-resolution may be seen as recovering missing high frequency content of frames after performing a simple interpolation on the LR frames. [F.Brandi (2008)] proposed an algorithm for recovering missing high-frequency contents of each block of LR frames. In their algorithm, high-frequency content of HR still images is extracted by convolving them with a special filter, then missing high frequency content of blocks of primarily interpolated LR frames are found by finding their best match in the HR still images.

Finding the true best match is not always possible due to occlusions and complex motions in the scene. As a remedy, [B. Song (2011)] used the same matching method in parallel with a learning based approach reserved for the cases where no relevant match could be found in the HR still images. Their proposed learning based method solves this problem by building class specific predictors learned based on the HR still images for predicting missing high-frequency contents of the LR frames. They assign a predictor to each class of LR blocks that estimates each pixel of associated HR block by performing a weighted averaging on the all pixels of LR block. Weights

are already defined for each class by using HR still images and their corresponding LR versions as learning set. However, associating a linear dependency between HR and LR patches without considering adjacent frames and degradation model leads to blurred restored HR frames.

The goal of this chapter is to effectively exploit all information including adjacent LR frames and HR still images for super-resolution. It is based on the idea of multi-frame super-resolution, and performing up-sampling task separately from de-blurring for easier implementation. We proposed a regularization function for de-blurring and de-noising.

The rest of this chapter is as follows. In section 3.2, a brief overview on the proposed algorithm is given. Section 3.3 and 3.4 cover up-sampling and de-blurring tasks, respectively. Section 3.5 describes employing Generalized Cross Validation (GCV) method for defining regularization parameter. Experimental results are presented in section 3.6. This chapter is concluded in section 3.7.

3.2 Overview of the Proposed Algorithm

Super-resolution reconstruction of each frame takes the reverse path of the degradation to obtain original HR frame. The degradation process includes camera PSF blurring, LR grid down-sampling, and contamination by noise. One possible mathematical model for degradation as described in the previous chapter is:

$$\boldsymbol{y} = \boldsymbol{DHx} + \boldsymbol{n} \tag{3.1}$$

where pixels of the current LR and HR frames are rearranged in lexicographic order and form $\boldsymbol{y}$ and $\boldsymbol{x}$ vectors. $\boldsymbol{D}$ and $\boldsymbol{H}$ are two matrices representing down-sampling and PSF blurring, respectively. $\boldsymbol{n}$ represents image sensor noise with variance equal to σ^2. As explained in the previous chapter, adjacent frames carry relevant information and are added to the problem formulation in multi-frame super-resolution approach. Briefly, multi-frame super-resolution process can be viewed and explained as two tasks:

1-Registering adjacent LR frames and finding sub-pixel motions between current frame and adjacent LR frames, and fusing them on the HR grid. The result is an up-sampled frame which is blurred and noisy as described in previous chapter. We call this up-sampling step.

2-De-noising and de-blurring of the up-sampled frame using blur model, and an image prior(*i.e.* regularization function)

Performance of the first step relies on the existence of motion in the scene. Motion can reveal lost points on the HR grid of one frame in its adjacent frames. As explained in the previous chapter, accuracy of the motion estimation and access to sufficient number of adjacent frames carrying relevant information have a great impact on the final quality of super-resolved frame.

As explained in the last section of the previous chapter, these tasks may be performed simultaneously which is the optimal case. But in our proposed super-resolution algorithm presented in this chapter, these tasks are performed separately. A block-diagram of different steps of process is depicted in Figure 3.1.

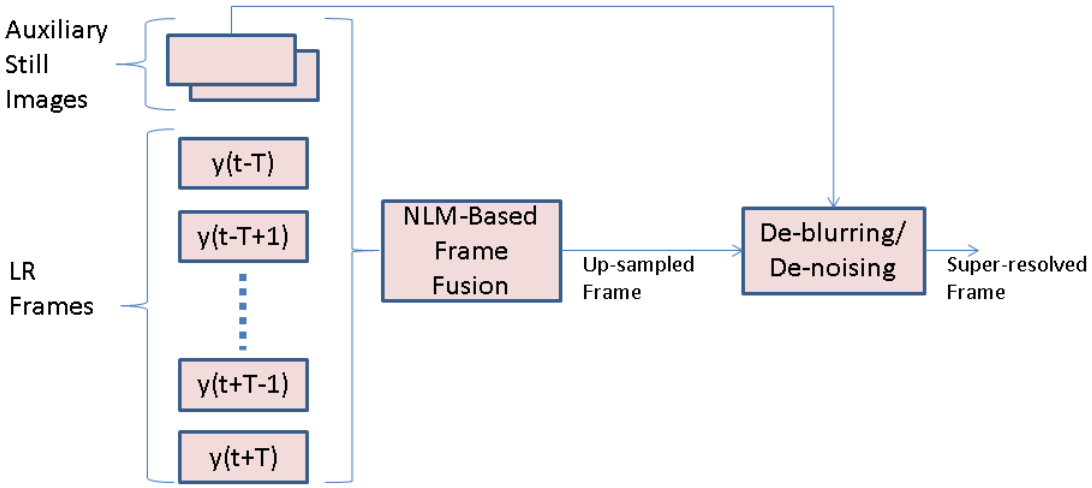

Figure 3.1: Block-diagram of proposed super-resolution algorithm consists of frame fusion and de-bluring/de-noising

3.3 Up-sampling : NLM-based Frame Fusion

As stated before, presence and estimation of sub-pixel motion between frames is not practical in the real world. Therefore, methods implicitly estimate sub-pixel motion have been proposed recently. One of the most effective method inspired by non-local means filter is proposed in [Protter *et al.* (2009)]. It suggests to generalize Non-local means de-noising approach [Antoni Buades, Bartomeu Coll, Jean-Michel Morel (2011)] for the multi-frame super-resolution problem. Before reviewing this method, principals of non-local means filter is discussed in this section.

Non-Local Means (NLM) filter and bi-lateral filter belong to the same family of de-noising methods. These methods are very successful as de-noising tools and are based on the fact that an image exhibits self-similarity. Therefore, each pixel can be de-noised by averaging all similar pixels in its neighborhood. Each neighborhood pixel contributes partially in the averaging process, according to a weigh associated

to it based on a similarity measure. The formulation for the restored pixel is:

$$\hat{\boldsymbol{x}}(k,l) = \frac{\sum\limits_{(i,j)\in\Omega_{(k,l)}} w(k,l,i,j)\boldsymbol{y}(i,j)}{\sum\limits_{(i,j)\in\Omega_{(k,l)}} w(k,l,i,j)} \tag{3.2}$$

where $\Omega_{(k,l)}$ stands for neighborhood set of the pixel located at (k,l), and $w(k,l,i,j)$ stands for the weight associated to the pixel located at (i,j) to contribute in construction of pixel located at (k,l). $w(k,l,i,j)$ can be represented using a general equation:

$$w(k,l,i,j) = \exp(-\left\|\mathbf{R}_{(k,l)}\boldsymbol{x} - \mathbf{R}_{(i,j)}\boldsymbol{y}\right\|_2^2 / 2\delta^2).\Phi(\|(k-i,l-j)\|_2) \tag{3.3}$$

where Φ is a decreasing function, and $\mathbf{R}_{(m,n)}$ is an operator extracting a patch centered at the location (m,n). δ is a parameter controls relative contribution of candidate pixels in the neighborhood of pixel located at (k,l). The difference between NLM and bi-lateral filter lies in the patch size. The patch size for bi-lateral filter is only one pixel, while the patch for NLM can be any size. It enables NLM filter to measure the similarity of two pixels more accurately. It is worth mentioning that the way these weights are calculated is intuitive and exponential form is first suggested by [Antoni Buades, Bartomeu Coll, Jean-Michel Morel (2011)]. The exponential form relation between weight and negated sum of absolute differences, eliminates non-related patches and penalizes differences severely. [Protter *et al.* (2009)] uses the idea behind this method to perform motion estimation and frame fusion and to up-sample a video frame. De-blurring and de-noising of the resulted up-sampled frame is then carried out using BTV regularization function in the algorithm proposed in [Protter

et al. (2009)].

[Protter *et al.* (2009)] formulated motion estimation and frame fusion as an energy function minimization and solved it to reach a closed form solution for the frame fusion. Before explaining the derivation of the non-local means based frame fusion method, we want to take a deeper look at the origin of non-local means filter. Non-local means de-noising filter can be derived by minimizing a cost function. Getting familiar with this derivation is useful to understand how it works, and how it may used for the frame fusion in the multi-frame super-resolution process.

This cost function imposes each pixel of the restored image to have a close value to other pixels with similar patches in its vicinity. Consider $\boldsymbol{x}$ is the original frame and $\boldsymbol{y}$ is its noisy version. The NLM de-noising filter cost function is as follows:

$$e_{NLM}(\boldsymbol{x}) = \sum_{(k,l)} \sum_{(i,j)\in\Omega_{(k,l)}} w(k,l,i,j) \left\| \mathbf{P}_{(k,l)}\boldsymbol{x} - \mathbf{P}_{(i,j)}\boldsymbol{x} \right\|_2^2$$

Where $\Omega_{(k,l)}$ is the set of all candidate pixels weighted inside the search region for the pixel located at (k,l). $\mathbf{P}_{(m,n)}$ is an operator taking out the pixel placed at location (m,n). For instance, $\mathbf{P}_{(m,n)}\boldsymbol{x} = \boldsymbol{x}(m,n)$. This cost function imposes the similarity between pixels with the similar patches. By minimizing this cost function iteratively, and substituting $\boldsymbol{x}^0 = \boldsymbol{y}$, the first iteration leads to the NLM filter described by (3.2). More details about this derivation is available in [Protter *et al.* (2009)]. Now, consider we want to use the same idea for up-sampling a LR frame by fusing all adjacent frames. Assume T LR frames are fused for super-resolving the current frame which is called $\boldsymbol{x}$. The unknown we are looking for in the up-sampling step is $\boldsymbol{z} = \boldsymbol{H}\boldsymbol{x}$ which is the result of fusion of all LR frames. Fusion of LR frames can only lead to $\boldsymbol{z} = \boldsymbol{H}\boldsymbol{x}$ as pixels of LR frames are already affected by $\boldsymbol{H}$. Although fusion and restoration tasks

(de-blurring and de-noising) can be done simultaneously theoretically by one cost function, for the sake of implementation simplicity, it is suggested to perform them separately [Protter *et al.* (2009)]. Now, we focus on the up-sampling task to achieve $\boldsymbol{z}$ by fusion of LR frames. We want to explain up-sampling task as a process similar to de-noising to exploit NLM de-noising filter for it. The first difference between up-sampling and de-noising is that the size of frame changes during up-sampling, and we want to fuse LR frames to get a HR frame. To handle this problem, [Protter *et al.* (2009)] suggests to first map pixels of the all LR frames on their HR grid properly. Assume up-sampling is performed for a factor of r. Therefore, we can fill $\frac{1}{r^2}$ pixel locations on the HR grid of each LR frame by just moving its pixels to the proper location. This location depends on the fact that how we modeled camera PSF and LR frame formation by blurring and down-sampling matrices $\boldsymbol{H}$ and $\boldsymbol{D}$. For example, if up-sampling is performed for a factor of 2 and each LR pixel is the result of averaging over all 4 neighboring pixels, we may construct $\boldsymbol{H}$ based on the following blurring window:

$$\frac{1}{4}\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

and down-sampling matrix as a matrix which picks pixels at odd indexed rows and columns, and eliminates the rest. Based on this structure, during mapping each LR frame's pixels on its HR grid we should put (i, j) pixel of LR frame on the $(2(i-1)+1, 2(j-1)+1)$ pixel location of the HR grid. After this step, we will have $\frac{r^2-1}{r^2}$ holes on the HR grid without any value. We primarily fill this locations by performing a simple interpolation on the existing pixels. Now, the problem setup is more similar to a de-noising problem, and the goal is to estimate all pixels on the

HR grid of current frame (including interpolated and non-interpolated pixels) using original (non-interpolated) pixels of all adjacent frames. Interpolated pixels in the adjacent frames are not considering as candidate pixels, since they do not carry any new information and are calculated based on the original pixels. The following cost function performs this task for us:

$$e_f(\boldsymbol{z}) = \sum_{(k,l)\in\Omega_{\boldsymbol{z}}} \sum_{t\in[1\ldots T]} \sum_{(i,j)\in\Omega^t_{(k,l)}} w(k,l,i,j,t) \left\| \mathbf{P}_{(k,l)}\boldsymbol{z} - \mathbf{P}_{(i,j)}\tilde{\boldsymbol{y}}_t \right\|_2^2$$

$$\hat{\boldsymbol{z}} = \min_z \{e_f(\boldsymbol{z})\}$$

where $\tilde{\boldsymbol{y}}_t$ is the t^{th} LR frame involved in the up-sampling process after it is mapped and interpolated on the HR grid. $\Omega^t{}_{(k,l)}$ is the set of all candidate pixels among non-interpolated pixels, inside the search region for the pixel located at (k,l) in its adjacent frame $\hat{\boldsymbol{y}}_t$. $\Omega^{\boldsymbol{z}}$ represents all pixels inside frame $\boldsymbol{z}$. Assuming$\boldsymbol{P}_{(m,n)}$ extracts only the pixel at location (m,n), minimizing this cost function in the same way we explained for minimizing cost function of NLM filter leads to the following close form expression for reconstructing pixel located at (k,l) in the current frame :

$$\hat{\boldsymbol{z}}(k,l) = \frac{\sum_{t=1}^{T} \sum_{(i,j)\in\Omega^t{}_{(k,l)}} w(k,l,i,j,t)\tilde{\boldsymbol{y}}_t(i,j)}{\sum_{i=1}^{T} \sum_{(i,j)\in\Omega^t{}_{(k,l)}} w(k,l,i,j,t)} \tag{3.4}$$

where $\hat{\boldsymbol{z}}(k,l)$ is the estimation for $\boldsymbol{z}(k,l)$. Therefore, all candidate pixels found inside all primarily up-scaled LR frames contribute to the reconstruction of the pixel located at (k,l) in the current frame by the above formula. The weight associated to the candidate pixel located at (i,j) in an adjacent frames $\tilde{\boldsymbol{y}}_t$ is calculated similar to NLM

filter by an exponential term as follows:

$$w(k,l,i,j,t) = \exp(-\left\|\mathbf{R}_{(k,l)}\tilde{\boldsymbol{z}} - \mathbf{R}_{(i,j)}\tilde{\boldsymbol{y}}_t\right\|_2^2 / 2\delta^2) \tag{3.5}$$

where $\tilde{\boldsymbol{z}}$ is an initial up-sampled version of current frame resulted from simple interpolation. $\mathbf{R}_{(m,n)}$ is an operator extracting the patch centered at the location (m,n). δ is the parameter controlling the relative contribution of different pixels based on their local similarity to the pixel located at (k,l).

The difference between the described frame fusion method and conventional explicit motion compensation based methods is the type of motion they employ. The NLM-based method tries to find motion of each pixel by many motion vectors in a fuzzy manner. Conventional methods determine just one motion vector per adjacent frame for each pixel. In fact, there is no guarantee that pixels of the adjacent frames can be exactly mapped on the HR grid of current frame. In the NLM-based frame fusion method, instead of choosing the best motion vector, many motion vectors with different contributions are employed for each pixel (See figure 3.2). In this way, complex forms of motion can be handled since we are not looking for the "exact match". In other words, in NLM-based method averaging replaces the maximization, and this enhances performance effectively [Protter *et al.* (2009)]. There are two differences between the frame fusion method described in [Protter *et al.* (2009)] and the way we use it in our application. First, we have access to HR auxiliary frames, therefore we blur them intentionally using blurring function $\boldsymbol{H}$ and use them like other adjacent frames during this process. Second, we define search region in another manner. The method described in [Protter *et al.* (2009)] weights all pixels located inside the search region in the adjacent frames while most of the pixels are associated with negligible

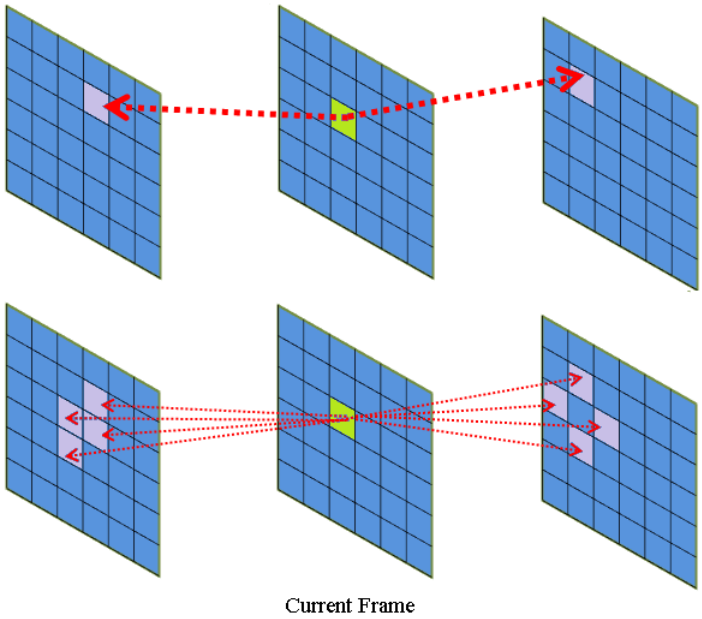

Figure 3.2: Conventional motion estimation (up) vs. Fuzzy motion estimation (down) for frame fusion (each section represents one pixel)

weights. Therefore, we employed another method to effectively search the most probable regions and reduce complexity of the algorithm. Since not every pixels in the search region carry relevant information, we adopt a pattern of locations to search for good candidates. The pattern we used is the famous pattern used in "Diamond Search" (DS) [Zhu and Ma (2000)]. There is a substantial difference between how DS is employed in our algorithm and how is it used for conventional motion estimation purposes. In conventional motion estimation applications, DS is employed for finding the best match. In the proposed algorithm, DS is used to find the region that best matches probably exist, and all of pixels which DS examines to lead to the best matches are weighted as candidate pixels. As a result, $\Omega^t{}_{(k,l)}$ is not set of all pixels inside the search area, and is only pixels which examined by the DS algorithm.

At this point, all discarded samples on the HR grid is estimated using adjacent

LR frames and $\hat{\boldsymbol{z}}$ as an estimate of $\boldsymbol{z} = \boldsymbol{Hx}$ is obtained.

Now, restoration should be performed to remove low-pass filtering effect of image sensor's PSF and noise. For this task, it is important to take it into account that noise distribution is different for $\hat{\boldsymbol{z}}$ compared to $\boldsymbol{y}_t$. Image sensor noise has already contaminated samples of $\boldsymbol{y}_t$, therefore; $\hat{\boldsymbol{z}}$ which is related to it by (3.4) is also contaminated by this noise. Moreover, $\hat{\boldsymbol{z}}$ in (3.4) is an estimate of the true sample of $\boldsymbol{Hx}$, and this estimation is associated with error.

According to (3.4), samples of $\hat{\boldsymbol{z}}$ are weighted averages of samples of $\boldsymbol{y}_t$. In terms of noise, this averaging changes the noise energy of $\hat{\boldsymbol{z}}$, compared to $\boldsymbol{y}_t$. Although Gaussian Identically Independently Distributed (i.i.d.) assumption for the noise contaminating $\boldsymbol{y}$ in (3.1) is simplifying and justifiable, we should define properties of the noise contaminating $\hat{\boldsymbol{z}}$ more carefully. Weighted averaging process over samples of $\boldsymbol{y}_t$ does not change noise distribution of the resulted sample, because summation of samples of some Gaussian distributions is also Gaussian. $\hat{\boldsymbol{z}}$ samples' covariance matrix should be calculated with respect to the covariance matrix of $\boldsymbol{y}$ samples. Considering (3.4) variance of noise contaminating $\hat{\boldsymbol{z}}(k,l)$ is as follows:

$$\sigma^2_{(k,l)} = \frac{\sum_{t=1}^{T} \sum_{(i,j)\in\Omega^t_{(k,l)}} w^2(k,l,i,j,t)}{\left(\sum_{t=1}^{T} \sum_{(i,j)\in\Omega^t_{(k,l)}} w(k,l,i,j,t)\right)^2}\sigma^2 = \frac{\left\|W^{(k,l)}\right\|_2^2}{\left\|W^{(k,l)}\right\|_1^2}\sigma^2 \tag{3.6}$$

where σ^2 is variance of noise contaminating $\boldsymbol{y}$, and $W^{(k,l)}$ is the set of all weights found for pixel located at (k,l). Obviously, this value is smaller than σ^2 and up-scaling step itself reduces the effect of image sensor noise. But as stated before, image sensor noise is not the only noise affecting $\hat{\boldsymbol{z}}$ contrary to $\boldsymbol{y}_t$. $\hat{\boldsymbol{z}}$ is supposed to be an estimate for

$\boldsymbol{Hx}$. The error of this estimation depends on the type of the motion associated with each pixel. Each pixel of $\hat{\boldsymbol{z}}$ is obtained by taking weighted average of similar pixels in all adjacent LR frames with respect to (3.4). This estimate can lead to a value substantially different from the exact value for a pixel if it is occluded in the adjacent frames and there is not enough candidates for it. Therefore, the error associates with the estimation is separate from noise of image sensor which contaminates all LR frames and consequently the $\hat{\boldsymbol{z}}$.

It is worth mentioning, similar problem exists wherever any form of motion estimation is used in an algorithm, and it is not easy to determine the error of motion estimation in these cases. For example, explicit motion estimation is used to find $\boldsymbol{F}_t$ matrices for conventional frame fusion which is described in the previous chapter. Each element of $\boldsymbol{F}_t$ shows how a pixel has changed its position in a frame, but we do not have a measure for its correctness to include it in our formulation. Hereafter, we assume that the total effect of the aforementioned sources which contaminate samples of $\hat{\boldsymbol{z}}$ is an i.i.d noise and therefore, Σ is as follows

$$\Sigma = \begin{pmatrix} 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{pmatrix} \sigma_z^2 = \boldsymbol{I}\sigma_z^2$$

where σ_z is the total effect of the variance of noise and estimation for pixels of $\hat{\boldsymbol{z}}$. At the next step, de-blurring and de-noising should be performed to remove low-pass filtering effect of image sensor's PSF and noise.

3.4 De-blurring

In previous section we obtained $\hat{\boldsymbol{z}}$ as an estimate to $\boldsymbol{z} = \boldsymbol{H}\boldsymbol{x}$. De-blurring and de-noising process is required to estimate $\boldsymbol{x}$ from $\hat{\boldsymbol{z}}$. For simplicity we will use $\boldsymbol{z}$ instead of $\hat{\boldsymbol{z}}$ for representing the up-sampled frame obtained from previous section in the rest of this chapter. This process can be formulated as a regularized reconstruction approach based on the reasons discussed about regularization functions in the previous chapter:

$$\begin{aligned} e(\boldsymbol{x}) &= (\boldsymbol{H}\boldsymbol{x} - \boldsymbol{z})^T(\boldsymbol{H}\boldsymbol{x} - \boldsymbol{z}) + \gamma\Psi(\boldsymbol{x}) \\ \hat{\boldsymbol{x}} &= \min_x\{e(\boldsymbol{x})\} \end{aligned} \tag{3.7}$$

where $\Psi(\boldsymbol{x})$ is generally the regularization function. Conventional regularization functions impose general features on restored image such as smoothness or sparsity among transform coefficients. Using a regularization function specified by samples from the same scene can mitigate our need for adhering to a general characteristics such as smoothness. Inspired by the energy function used for NLM filter derivation, we propose a regularization function, ψ_r, of the form

$$\begin{aligned} \psi_r(\boldsymbol{x}) &= \sum_{t=1}^{2} \sum_{(k,l)\in\Omega_x} \bar{w}_{(k,l)}^{-1} \sum_{(i,j)\in\Omega_{(k,l)}^t} w(k,l,i,j,t) \left\| \mathbf{P}_{(k,l)}\boldsymbol{x} - \mathbf{P}_{(i,j)}\boldsymbol{x}_t^r \right\|_2^2 \\ \bar{w}_{(k,l)} &= \sum_{t=1}^{2} \sum_{(i,j)\in\Omega_{(k,l)}} w(k,l,i,j,t) \end{aligned} \tag{3.8}$$

to play the role of $\Psi(\boldsymbol{x})$ in (3.7). $\boldsymbol{x}_t^r$ represents previous and next HR still images for $t = 1$ and $t = 2$ respectively. Next and previous HR still image are the closest HR still images to the LR frame being reconstructed. We do not want to include other HR still images as they are far from the current LR frame and thus have negligible correlation. $\Omega_{(k,l)}^t$ is the set of candidate pixels found in the t^{th} auxiliary HR image

for pixel located at (k, l), and Ω_x represents all pixels of $\boldsymbol{x}$. Similar to previous section, candidate pixels are chosen using a method based on diamond search (DS) algorithm. As can be seen, when $w(k, l, i, j, t)$ is higher the difference between pixels of the frame being reconstructed and HR still images leads to higher cost, and the difference should be penalized more. The above definition for regularization function, reduces the difference between pixels of the reconstructed frame and pixels of the HR still images which have similar patches. $\bar{w}_{(k,l)}^{-1}$ is used as a coefficient behind the summation of all costs associated with the pixel located at (k, l) to normalize weights of all candidate pixels associated with that pixel. This way reconstructed frame reflects patterns of the HR still images when available (*i.e.* when $w(k, l, i, j, t)$ is not negligible).

One preliminary and straight-forward but not practical way for calculating weights $w(k, l, i, j, t)$ for this regularization function is as follows:

$$w(k, l, i, j, t) = \exp(-\left\|\mathbf{R}_{(k,l)}\boldsymbol{x} - \mathbf{R}_{(i,j)}\boldsymbol{x}_t^r\right\|_2^2 / 2\delta^2) \tag{3.9}$$

Using the above formula, we can measure similarities between restored frame and the auxiliary HR image. But, this formulation renders the derivative of (3.7) very complicated because it contains the variable $\boldsymbol{x}$. Moreover, if we want to solve (3.7) iteratively, we should update weights at each iteration and it severely increases computational complexity. We prefer to calculate these weights independent of $\boldsymbol{x}$. By a subtle change in this formula, we can still measure the similarities between current frame and auxiliary HR image independent of $\boldsymbol{x}$. The solution is to measure similarities between $\boldsymbol{z}$ which has been obtained already by the up-sampling process and

blurred versions of auxiliary HR still images in hand:

$$w(k,l,i,j,t) = \exp(-\left\|\mathbf{R}_{(k,l)}\boldsymbol{z} - \mathbf{R}_{(i,j)}\boldsymbol{H}\boldsymbol{x}_t^r\right\|_2^2 / 2\delta^2) \tag{3.10}$$

Although $\boldsymbol{z}$ is blurred and we want to de-blur it, it still reveals the similarities between unknown de-blurred frame $\boldsymbol{x}$ and the auxiliary HR images when we compare it to the intentionally blurred version of those images. This approximation is sub-optimal and its effectiveness depends on estimation accuracy of $\boldsymbol{z}$ from the previous step. Now, we can proceed with solving (3.7) using proposed regularization function in (3.8). To minimize the cost function in (3.7), we need to take its derivative with respect to $\boldsymbol{x}$ and equate it to zero:

$$\begin{aligned}
&\tfrac{de(\boldsymbol{x})}{d\boldsymbol{x}} = 2\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{x} - 2\boldsymbol{H}^T\boldsymbol{z} + \\
&2\gamma\sum_{t=1}^{2}\sum_{(k,l)\in\Omega_x}\bar{w}_{(k,l)}^{-1}\sum_{(i,j)\in\Omega_{(k,l)}} w(k,l,i,j,t)\mathbf{P}_{(k,l)}^T\mathbf{P}_{(k,l)}\boldsymbol{x} - \\
&2\gamma\sum_{t=1}^{2}\sum_{(k,l)\in\Omega_x}\bar{w}_{(k,l)}^{-1}\sum_{(i,j)\in\Omega_{(k,l)}} w(k,l,i,j,t)\mathbf{P}_{(k,l)}^T\mathbf{P}_{(i,j)}\boldsymbol{x}_t^r = \mathbf{0}
\end{aligned} \tag{3.11}$$

considering:

$$\sum_{t=1}^{2}\sum_{(k,l)\in\Omega_x}\bar{w}_{(k,l)}^{-1}\sum_{(i,j)\in\Omega_{(k,l)}} w(k,l,i,j,t)\mathbf{P}_{(k,l)}^T\mathbf{P}_{(k,l)}\boldsymbol{x} = \boldsymbol{x} \tag{3.12}$$

and:

$$\boldsymbol{\Delta}_r \triangleq \sum_{t=1}^{2}\sum_{(k,l)\in\Omega_x}\bar{w}_{(k,l)}^{-1}\sum_{(i,j)\in\Omega_{(k,l)}} w(k,l,i,j,t)\mathbf{P}_{(k,l)}^T\mathbf{P}_{(i,j)}\boldsymbol{x}_t^r \tag{3.13}$$

therefore, we can equate $\frac{de(\boldsymbol{x})}{d\boldsymbol{x}}$ to zero and solve the equation for $\boldsymbol{x}$ as follows,

$$(\boldsymbol{H}^T\boldsymbol{H} + \gamma\boldsymbol{I})\boldsymbol{x} - \boldsymbol{H}^T\boldsymbol{z} - \gamma\Delta_r = \mathbf{0} \tag{3.14}$$

$$\hat{\boldsymbol{x}} = (\boldsymbol{H}^T \boldsymbol{H} + \gamma \boldsymbol{I})^{-1} (\boldsymbol{H}^T \boldsymbol{z} + \gamma \Delta_r) \tag{3.15}$$

where $\mathbf{P}^T_{(k,l)} w(k,l,i,j,t)$ builds a vector with one non-zero element equal to $w(k,l,i,j,t)$ placed at the location (k,l). To understand this solution, Δ_r should be explained. Taking a look at the (3.13) describing Δ_r we see each pixel of Δ_r is a weighted average of pixels of closest auxiliary still images shown by $\boldsymbol{x}^r_1$ and $\boldsymbol{x}^r_2$. This fact can be explained further as follows. $\mathbf{P}_{(i,j)} \boldsymbol{x}^r_t$ takes pixels of $\boldsymbol{x}^r_t$ located inside search region ($(i,j) \in \Omega_{(k,l)}$) out. Each of them is multiplied by the associated normalized weight $\bar{w}^{-1}_{(k,l)} w(k,l,i,j,t)$, then all results are added and put at the location (k,l) of the Δ_r using $\mathbf{P}^T_{(k,l)}$ operator. In fact, Δ_r is a very good primary estimate of the final reconstructed frame and we call it $\hat{\boldsymbol{x}}_{reg}$:

$$\hat{\boldsymbol{x}}_{reg} = \Delta_r. \tag{3.16}$$

This estimates can be easily derived by separately minimizing $\psi_r(\boldsymbol{x})$ for $\boldsymbol{x}$. It means that similarity to $\hat{\boldsymbol{x}}_{reg}$ is what the proposed regularization function imposes on the final frame. $\hat{\boldsymbol{x}}_{reg}$ is the auxiliary HR still images registered with respect to the current LR frame. Therefore, we see the proposed regularization function penalizes the difference between restored frame and $\hat{\boldsymbol{x}}_{reg}$. Although $\hat{\boldsymbol{x}}_{reg}$ may be considered as a primary solution for super-resolution problem itself, it is degraded in the areas which is occluded and the algorithm cannot find any match for them in the closest auxiliary HR still images. Note that LR frame and degradation process (matrix $\boldsymbol{H}$) are completely ignored in the $\hat{\boldsymbol{x}}_{reg}$, and it is only based on HR still images. Understanding the impact of the proposed regularization function on the final restored frame, we

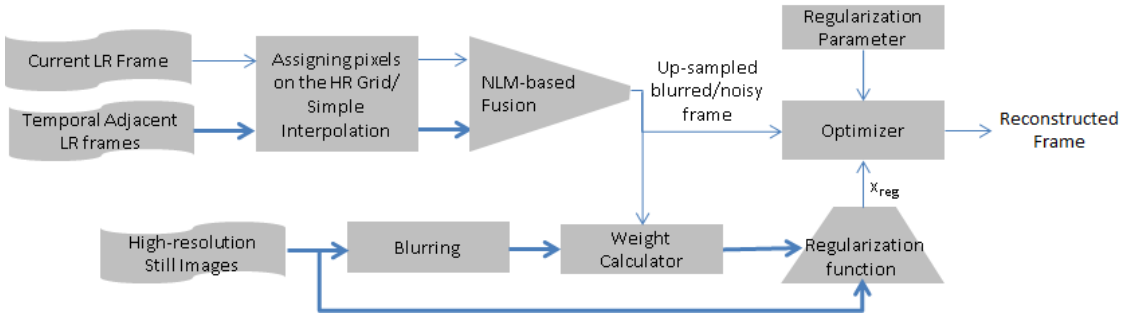

Figure 3.3: Full diagram of the proposed method

may re-formulate the problem as follows:

$$e(\boldsymbol{x}) = (\boldsymbol{Hx} - \boldsymbol{z})^T(\boldsymbol{Hx} - \boldsymbol{z}) + \gamma(\boldsymbol{x} - \Delta_r)^T(\boldsymbol{x} - \Delta_r)$$
$$\hat{\boldsymbol{x}} = \min_x\{e(\boldsymbol{x})\} \tag{3.17}$$

Solving this minimization problem we achieve the same answer obtained in (3.15) and is just representing the problem in another from which is called constrained reconstruction approach, and the constraint is Δ_r, defined based on the auxiliary still images. As stated in previous paragraph, Δ_r is sharp but degraded in some regions where the current frame do not have good matches in the auxiliary HR still images. In the occluded regions, first term of the cost function which does not depend on the auxiliary still images tends to push the optimize answer to a more accurate value by incorporating $\boldsymbol{z}$. The experimental results show that occluded regions are handled effectively in this way. Figure 3.3 illustrates a complete diagram of the proposed method.

3.5 Generalized Cross Validation Method for Defining Regularization Parameter

We showed how the recovered frame should look like auxiliary HR still images through the regularization function introduced in previous section. But, it is not determined yet to what extent the restored frame should obey regularization function and to what extent it should be loyal to observed data from LR frames. It is very important to set this trade-off in a manner to obtain the best possible result. γ is the regularization parameter controlling this trade-off and increasing and decreasing it increases the effect of auxiliary HR still images and LR frames, respectively. Knowing the noise covariance matrix associated with noisy observation, including ours, may facilitate defining this parameter but in most applications noise covariance matrix is not known. Finding the optimal value for regularization parameter has been discussed in details for conventional regularization functions and many algorithms has been proposed for that. L-curve [Hansen (2000)] and Generalized Cross Validation (GCV) are the most popular methods. L-curve method associates with calculating of curvature of a plot, and is computationally extensive. In this work we exploit GCV to find the regularization parameter γ.

3.5.1 Cross Validation and Generalized Cross Validation

The idea of cross validation is used to evaluate an assumption on a set of observations. Based on this idea, observation set is divided into two subsets: estimation and validation subsets. Estimation subset is used to estimate some parameters of

observation data based on a number of assumptions. Validation subset is used to assess validity of this estimate and consequently the validity of assumptions associates with it. The question is that which samples should be chosen as estimation set and which samples should be used as validation set. If the noise variance was known, the problem would be much easier to solve, but in most cases and also in our problem it is not known. [Craven and Wahba (1979),Reeves (1991)] used GCV for determining γ when regularization function is Tikhonove. We will use the same methodology to solve our problem. Generalized Cross Validation is based on the same idea but each observation sample is used at least one time as validation sample and one time as estimation sample during the procedure. To this end, cross validation is performed on the observation data set S times, where S is equal to the number of observation samples, each time one sample forms validation subset and the rest of samples form the estimation subset. The validity of assumption then is defined by averaging over all trials. Assume we want to define regularization parameter γ in (3.7) when regularization function is the proposed regularization function, $\psi_r(\boldsymbol{x})$. The minimizer of this cost function is (3.15). We can re-write this solution as a function of γ:

$$\hat{\boldsymbol{x}}(\gamma) = (\boldsymbol{H}^T\boldsymbol{H} + \gamma\boldsymbol{I})^{-1}(\boldsymbol{H}^T\boldsymbol{z} + \gamma\Delta_r) \tag{3.18}$$

To apply GCV for finding best γ, we need to find optimal γ that minimizes the following expression:

$$e(\gamma) = (\boldsymbol{H}\boldsymbol{x} - \boldsymbol{z})^T(\boldsymbol{H}\boldsymbol{x} - \boldsymbol{z}) + \gamma\psi_r(\boldsymbol{x}) \tag{3.19}$$

To apply GCV, each time we find the value of this cost function by leaving out one sample of $\boldsymbol{z}$ and minimizing it through (3.15). Then, the leaved out sample is used for validation of the obtained γ. Consider i^{th} sample of $\boldsymbol{z}$ have been left out, therefore the cost function that should be minimized as follows:

$$e_{i^-}(\gamma) = (\boldsymbol{H}_{i^-}\boldsymbol{x} - \boldsymbol{z}_{i^-})^T(\boldsymbol{H}_{i^-}\boldsymbol{x} - \boldsymbol{z}_{i^-}) + \gamma\psi_r(\boldsymbol{x}) \tag{3.20}$$

where $\boldsymbol{z}_{i^-}$ is the vector $\boldsymbol{z}$ when its i^{th} component are removed. Similarly, $\boldsymbol{H}_{i^-}$ is $\boldsymbol{H}$ when its i^{th} row is removed. The minimizer of new cost function in (3.20) is $\hat{\boldsymbol{x}}_{i^-}(\gamma)$ as follows:

$$\hat{\boldsymbol{x}}_{i^-}(\gamma) = (\boldsymbol{H}_{i^-}^T\boldsymbol{H}_{i^-} + \gamma\boldsymbol{I})^{-1}(\boldsymbol{H}_{i^-}^T z_{i^-} + \gamma\Delta_r) \tag{3.21}$$

This estimation is obtained based on a certain assumption for value of the parameter γ. Validation of this estimation is assessed by exploiting the left out sample, z_i.

$$v_i(\gamma) = (\boldsymbol{h}_i\hat{\boldsymbol{x}}_{i^-}(\gamma) - z_i)^2 \tag{3.22}$$

where $\boldsymbol{h}_i$, and z_i are i^{th} row and element of matrix $\boldsymbol{H}$ and $\boldsymbol{z}$, respectively. $v_i(\gamma)$ should be assessed for all elements of z and final assessment for any value of γ is obtained by averaging $v_i(\gamma)$ over i:

$$V(\gamma) = \frac{1}{S}\sum_{i=1}^{S} v_i(\gamma) \tag{3.23}$$

where S is equal to $M \times N$ and is the length of vector $\boldsymbol{z}$. To determine $V(\gamma)$ we start with $v_i(\gamma)$ as follows:

$$v_i(\gamma) = (\boldsymbol{h}_i\hat{\boldsymbol{x}}_{i^-}(\gamma) - z_i)^2$$

$$\boldsymbol{h}_i\hat{\boldsymbol{x}}_{i^-}(\gamma) = \boldsymbol{h}_i(\boldsymbol{H}_{i^-}^T\boldsymbol{H}_{i^-} + \gamma\boldsymbol{I})^{-1}(\boldsymbol{H}_{i^-}^T\boldsymbol{z}_{i^-} + \gamma\Delta_r)$$

considering

$$\boldsymbol{H}_{i^-}^T \boldsymbol{z}_{i^-} = \boldsymbol{H}^T \boldsymbol{z} - \boldsymbol{h}_i^T z_i$$

$\boldsymbol{h}_i \hat{\boldsymbol{x}}_{i^-}(\gamma)$ may be re-written as follows:

$$\boldsymbol{h}_i \hat{\boldsymbol{x}}_{i^-}(\gamma) = \boldsymbol{h}_i (\boldsymbol{H}_{i^-}^T \boldsymbol{H}_{i^-} + \gamma \boldsymbol{I})^{-1} (\boldsymbol{H}^T \boldsymbol{z} - \boldsymbol{h}_i^T z_i + \gamma \Delta_r)$$

therefore;

$$(\boldsymbol{h}_i \hat{\boldsymbol{x}}_{i^-}(\gamma) - z_i) = \boldsymbol{h}_i (\boldsymbol{H}_{i^-}^T \boldsymbol{H}_{i^-} + \gamma \boldsymbol{I})^{-1} (\boldsymbol{H}^T \boldsymbol{z} + \gamma \Delta_r) - [\boldsymbol{h}_i (\boldsymbol{H}_{i^-}^T \boldsymbol{H}_{i^-} + \gamma \boldsymbol{I})^{-1} \boldsymbol{h}_i^T + 1] z_i$$

considering

$$(\boldsymbol{H}_{i^-}^T \boldsymbol{H}_{i^-} + \gamma \boldsymbol{I})^{-1} = \frac{1}{1 - \alpha_{ii}} (\boldsymbol{H}^T \boldsymbol{H} + \gamma \boldsymbol{I})^{-1}$$

where $\alpha_{ii}(\gamma)$ is (i, i) element of the following matrix:

$$\boldsymbol{A}(\gamma) = \boldsymbol{H} (\boldsymbol{H}^T \boldsymbol{H} + \gamma \boldsymbol{I})^{-1} \boldsymbol{H}^T$$

$(\boldsymbol{h}_i \hat{\boldsymbol{x}}_{i^-}(\gamma) - z_i)$ may be re-written as follows,

$$(\boldsymbol{h}_i \hat{\boldsymbol{x}}_{i^-}(\gamma) - z_i) = \frac{1}{1 - \alpha_{ii}(\gamma)} \boldsymbol{h}_i (\boldsymbol{H}^T \boldsymbol{H} + \gamma \boldsymbol{I})^{-1} (\boldsymbol{H}^T \boldsymbol{z} + \gamma \Delta_r) - (\frac{\boldsymbol{h}_i (\boldsymbol{H}^T \boldsymbol{H} + \gamma \boldsymbol{I})^{-1} \boldsymbol{h}_i^T}{1 - \alpha_{ii}(\gamma)} + 1) z_i$$

considering:

$$\boldsymbol{h}_i (\boldsymbol{H}^T \boldsymbol{H} + \gamma \boldsymbol{I})^{-1} \boldsymbol{h}_i^T = \alpha_{ii}(\gamma)$$

$v_i(\gamma)$ is simplified as follows:

$$v_i(\gamma) = [\frac{1}{1-\alpha_{ii}(\gamma)}\boldsymbol{h}_i(\boldsymbol{H}^T\boldsymbol{H}+\gamma\boldsymbol{I})^{-1}(\boldsymbol{H}^T\boldsymbol{z}+\gamma\Delta_r) - (\frac{z_i}{1-\alpha_{ii}(\gamma)})]^2$$

$$= [\frac{1}{1-\alpha_{ii}(\gamma)}\boldsymbol{h}_i(\boldsymbol{H}^T\boldsymbol{H}+\gamma\boldsymbol{I})^{-1}(\boldsymbol{H}^T\boldsymbol{z}+\gamma\Delta_r - z_i)]^2$$

$$= \frac{1}{(1-\alpha_{ii}(\gamma))^2}[\boldsymbol{a}_i(\gamma)\boldsymbol{z} - z_i + \gamma\boldsymbol{h}_i(\boldsymbol{H}^T\boldsymbol{H}+\gamma\boldsymbol{I})^{-1}\Delta_r]^2$$

where $\boldsymbol{a}_i(\gamma)$ is i^{th} row of matrix $\boldsymbol{A}(\gamma)$. Therefore, $V(\gamma)$ can be re-written as:

$$V(\gamma) = \frac{1}{S}\sum_{i=1}^{S} v_i(\gamma) = \frac{1}{S}\sum_{i=1}^{S}\frac{[(\boldsymbol{a}_i(\gamma)\boldsymbol{z} - z_i) + \gamma\boldsymbol{h}_i(\boldsymbol{H}^T\boldsymbol{H}+\gamma\boldsymbol{I})^{-1}\Delta_r]^2}{[1-\alpha_{ii}(\gamma)]^2}$$

$\alpha_{ii}(\gamma)$ is the same for all is because $\boldsymbol{H}$ is a circulant matrix. This fact is proved later in this chapter (See (3.27), and notes following it). Therefore, we can substitute $[1-\alpha_{ii}(\gamma)]^2$ by $[1-\frac{1}{S}\sum_{j=1}^{N}\alpha_{jj}(\gamma)]^2$. Therefore,

$$V(\gamma) = \frac{\frac{1}{S}\left\|[(\boldsymbol{A}(\gamma)-\boldsymbol{I})\boldsymbol{z} + \gamma\boldsymbol{H}(\boldsymbol{H}^T\boldsymbol{H}+\gamma\boldsymbol{I})^{-1}\Delta_r]\right\|^2}{[\frac{1}{S}tr(\boldsymbol{I}-\boldsymbol{A}(\gamma))]^2} \tag{3.24}$$

where $tr()$ represents trace of a matrix. This derivation may be simplified by representing it by the eigenvalue form as follows:

$$V(\gamma) = S\frac{\sum_{i=1}^{S}[\frac{\gamma}{|\rho_i|^2+\gamma}]^2\left|\zeta_i - \rho_i\delta_i\right|^2}{[\sum_{i=1}^{S}\frac{\gamma}{|\rho_i|^2+\gamma}]^2} \tag{3.25}$$

where ρ_i is an eigenvalue of matrix $\boldsymbol{H}$ for each i, and δ_i and ζ_i are Discrete Fourier Transform (DFT) coefficients of Δ_r and $\boldsymbol{z}$, respectively. Derivation of eigenvalue form

of $V(\gamma)$ is easy and straight-forward as follows. Considering matrix $\boldsymbol{H}$ is a "circulant matrix", the following decomposition holds for it:

$$\boldsymbol{H} = \boldsymbol{F}\hat{\boldsymbol{H}}\boldsymbol{F}^T \Leftrightarrow \quad \boldsymbol{H} \in \boldsymbol{Circ} \tag{3.26}$$

where $\boldsymbol{Circ}$ is the set of all circulant matrices. $\boldsymbol{F}$ is the matrix of eigenvectors and equals to Fourier matrix. Eigenvector matrix is orthonormal and the same for all circulant matrices,

$$\boldsymbol{F} = [\boldsymbol{f}_0, \boldsymbol{f}_1, \boldsymbol{f}_2, ...\boldsymbol{f}_m]$$

$$\boldsymbol{f}_j = (1, w_j, w_j^2, w_j^3, ..., w_j^{m-1})^T, w_j = exp(\frac{2\pi ij}{m}), m = S^2$$

$\hat{\boldsymbol{H}}$ is matrix of eigenvalues of $\boldsymbol{H}$ and is a diagonal matrix:

$$\hat{\boldsymbol{H}} = \begin{pmatrix} \rho_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \rho_m \end{pmatrix}$$

we can re-write A as follows:

$$\boldsymbol{A}(\gamma) = \boldsymbol{H}(\boldsymbol{H}^T\boldsymbol{H} + \gamma\boldsymbol{I})^{-1}\boldsymbol{H}^T =$$

$$\boldsymbol{F}\hat{\boldsymbol{H}}\boldsymbol{F}^T(\boldsymbol{F}\hat{\boldsymbol{H}}^T\hat{\boldsymbol{H}}\boldsymbol{F}^T + \gamma\boldsymbol{I})^{-1}\boldsymbol{F}\hat{\boldsymbol{H}}^T\boldsymbol{F}^T = \boldsymbol{F}\hat{\boldsymbol{H}}(\hat{\boldsymbol{H}}^T\hat{\boldsymbol{H}} + \gamma\boldsymbol{I})^{-1}\boldsymbol{H}^T\boldsymbol{F}^T$$

assuming

$$\Lambda = \hat{\boldsymbol{H}}(\hat{\boldsymbol{H}}^T\hat{\boldsymbol{H}} + \gamma\boldsymbol{I})^{-1}\boldsymbol{H}^T$$

Λ is a diagonal matrix with elements on the diagonal as follows:

$$\lambda_i = \frac{|\rho_i|^2}{|\rho_i|^2 + \gamma}$$

therefore, $\boldsymbol{A}(\gamma)$ is diagonalized as follows:

$$\boldsymbol{A}(\gamma) = \boldsymbol{F}\Lambda\boldsymbol{F}^T \tag{3.27}$$

Considering (3.27), $\boldsymbol{A}(\gamma)$ is diagonalized by the Fourier matrix $\boldsymbol{F}$ as the eigenvector matrix. Referring to (3.26), it can be inferred that $\boldsymbol{A}(\gamma)$ is a circulant matrix, and all elements on its diagonal are the same. We continue the derivation by re-writing $\boldsymbol{I} - \boldsymbol{A}(\gamma)$ based on what is obtained for $\boldsymbol{A}(\gamma)$ as follows:

$$\boldsymbol{I} - \boldsymbol{A}(\gamma) = \boldsymbol{F}\boldsymbol{F}^T - \boldsymbol{A}(\gamma) = \boldsymbol{F}\boldsymbol{F}^T - \boldsymbol{F}\hat{\boldsymbol{H}}(\hat{\boldsymbol{H}}^T\hat{\boldsymbol{H}} + \gamma\boldsymbol{I})^{-1}\boldsymbol{H}^T\boldsymbol{F}^T\boldsymbol{F}\boldsymbol{F}^T =$$

$$\boldsymbol{F}[\boldsymbol{I} - \hat{\boldsymbol{H}}(\hat{\boldsymbol{H}}^T\hat{\boldsymbol{H}} + \gamma\boldsymbol{I})^{-1}\boldsymbol{H}^T]\boldsymbol{F}^T$$

therefore, i^{th} element of vector $(\boldsymbol{I} - \boldsymbol{A}(\gamma))\boldsymbol{z}$ will be obtained as follows:

$$[1 - \frac{|\rho_i|^2}{|\rho_i|^2 + \gamma}]\zeta_i = [\frac{\gamma}{|\rho_i|^2 + \gamma}]\zeta_i$$

where $\zeta_i = \boldsymbol{F}^T\boldsymbol{z}$. Considering eigenvectors of $\boldsymbol{H}$ are basic function of Fourier transform, ζ_i are DFT coefficients of $\boldsymbol{z}$. Similarly, the second part of noimnator $\gamma\boldsymbol{H}(\boldsymbol{H}^T\boldsymbol{H} + \gamma\boldsymbol{I})^{-1}\Delta_r$ is a vector which its i^{th} element is as follows:

$$[\frac{|\rho_i|^2}{|\rho_i|^2 + \gamma}]\gamma\delta_i$$

where $\delta_i = \boldsymbol{F}^T \Delta_r$. Substituting achieved eigenvalue form expressions into (3.24) leads to derivation of (3.25).

3.6 Experiments

We examined the performance of proposed algorithm on four CIF and two $720p$ HD videos. HD videos are "City" and "Shields". The PSF was a 3×3 Gaussian blurring window with $std = 0.7$, and the noise variance σ^2 was equal to 1. All frames were blurred, contaminated by noise and down-sampled by a factor of 2. We kept one HR frame un-degraded for every T LR frames, to play the role of auxiliary HR still images. 9×9 patches are used for similarity evaluation and weight computation in both up-sampling and de-blurring steps. Diamond Search (DS) is used for finding potential candidate pixels within a 32×32 search region during up-sampling step. The parameter d was equal to 0.25 and 1 for up-sampling and de-blurring steps, respectively. γ, the regularization parameter, was set to 0.175 for all experiments.

First, we examine the γ found by GCV method by comparing it to the optimal γ. This experiment is conducted for restoration 15^{th} frame of "News" and "Mobile" sequence when 1^{st} and 30^{th} are HR frames, and noise variance of LR frames is equal to 1. Figure 3.4, and Figure 3.6 illustrate mean-square errors (MSE) of restored frame using (3.15) for different values of γ for "News", and "Mobile" video sequence, respectively. In addition, GCV values for the restored frames are re-scaled for better illustration and shown in these figures. As can be seen in this graphs, GCV cost function is minimized for the γ which is close to optimal regularization parameter which minimizes true MSE of restored frame. Figure 3.5, and Figure 3.7 are illustrating restored frame for different regularization parameters, γ, for "News" sequence and

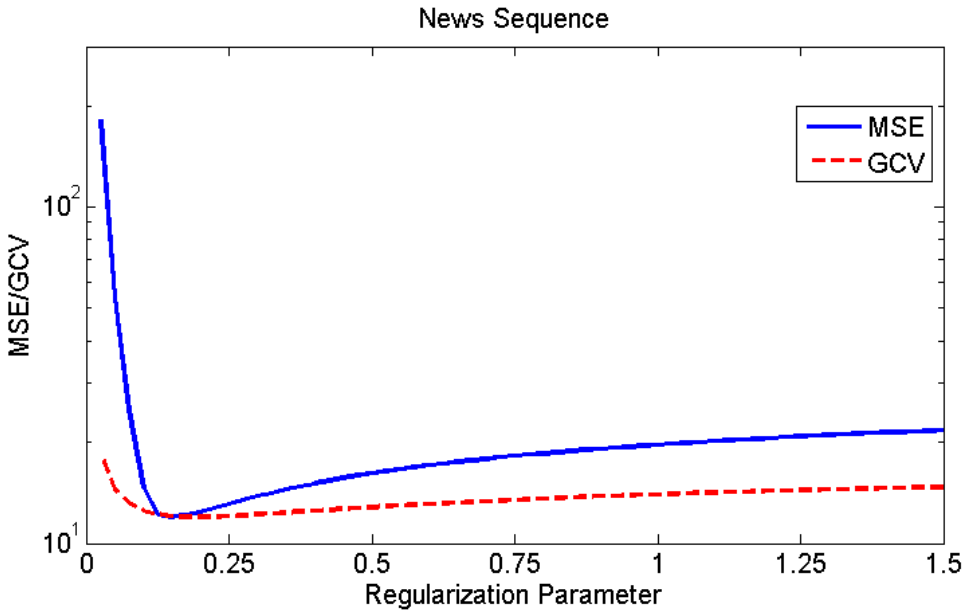

Figure 3.4: Finding optimal regularization parameter (γ) using GCV method for "News" video sequence.

"Mobile" sequence, respectively. It shows that small regularization parameter leads to noisy and undesirable restored frame. Very high regularization parameter $\gamma = 1.5$ leads to a frame which only depends on auxiliary HR images and some parts of the LR frame which can not be found in these HR images is lost for "News" sequence. γ equal to 0.175 and 0.1 are optimized regularization parameter found by minimizing GCV cost function for "News" and "Mobile" sequence, respectively and shows the best restoration quality among the other restorations.

Before comparison to other methods, we show effectiveness of regularized reconstruction approach compared to registration based approaches for exploiting the information of HR still images. In addition, we want to show how frames adjacent to auxiliary HR still images are restored with higher PSNR due to having more correlation with them. The interval between HR still images was 10 frames, and PSNR of

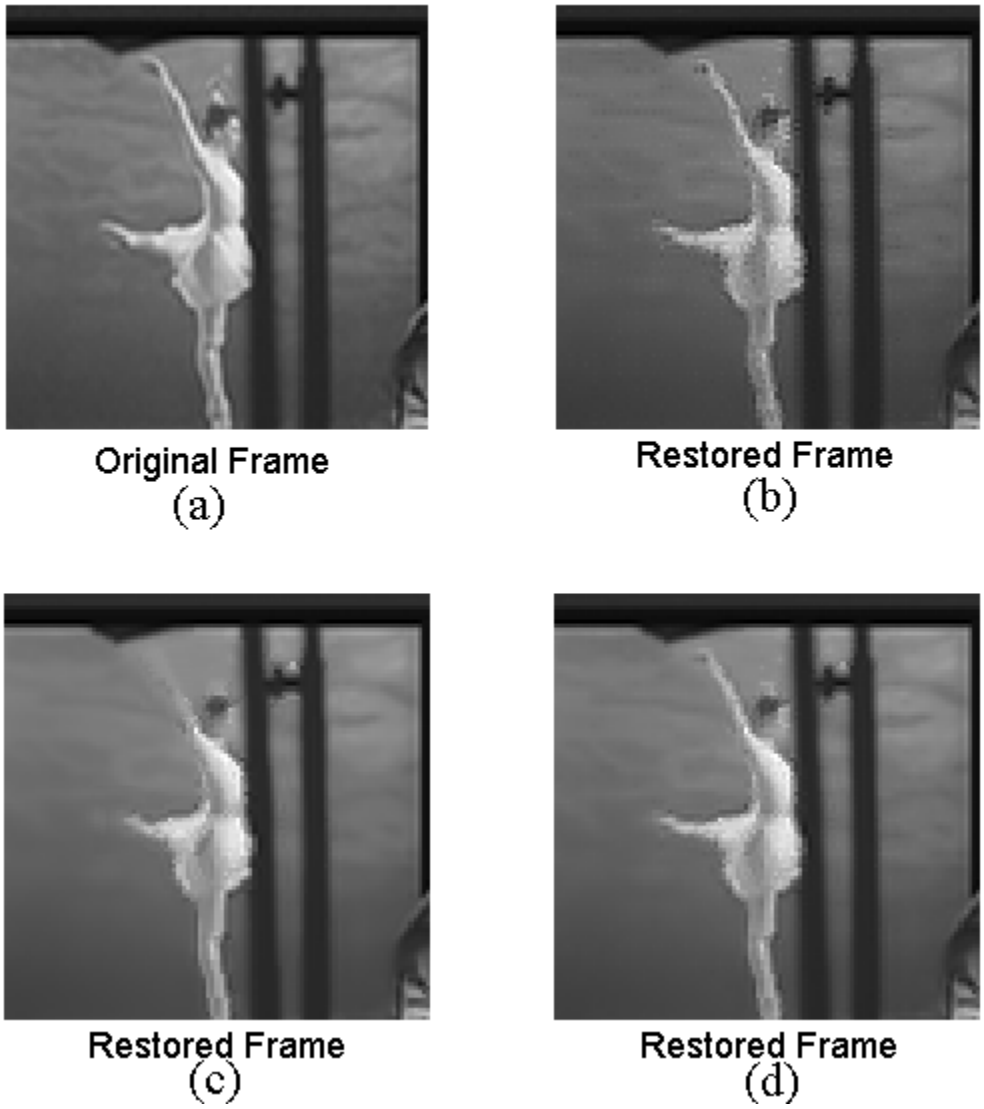

Figure 3.5: Restored frames using different regularization parameters for "News" sequence. (a) original image, (b) super-resolved by $\gamma = 0.025$, (c) super-resolved by $\gamma = 1.5$, (d) super-resolved by $\gamma = 0.175$

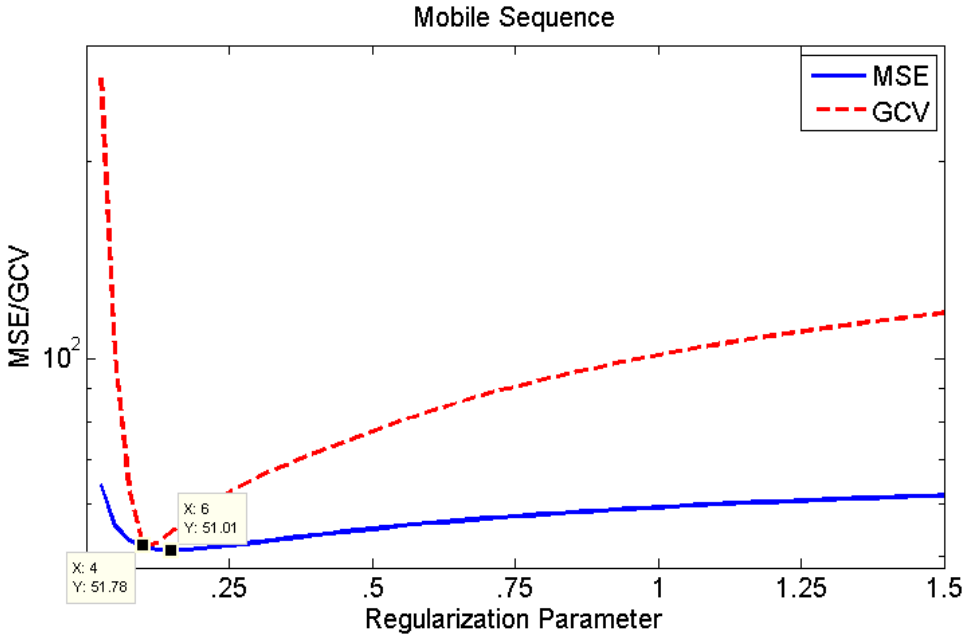

Figure 3.6: Finding optimal regularization parameter (γ) using GCV method for "Mobile" video sequence.

reconstructed frames inside two intervals is shown in the Figure 3.8 for the "Mobile" frame sequence. Results show that about 1 dB of PSNR is gained, when we use $\psi_r(\boldsymbol{x})$ as a regularization function in (3.15) instead of minimizing it separately for $\boldsymbol{x}$ to obtain $\hat{\boldsymbol{x}}_{reg}$ using equation (3.16). In addition, we saw that frames places at the middle of interval between two HR still images are restored with lower PSNR. This problem becomes severe if we increase this interval because that way, the time distance between HR still images and LR frames in the middle of interval increases and those frames will not reconstruct properly as their correlation with the auxiliary still images is decreased due to the motion. We conclude here that the interval between HR still images has a maximum for algorithm to be practical and it depends on the motion of the scene. As this interval is a factor defined by the camera and relates to its limitations, the proposed algorithm is suitable for videos with limited motion.

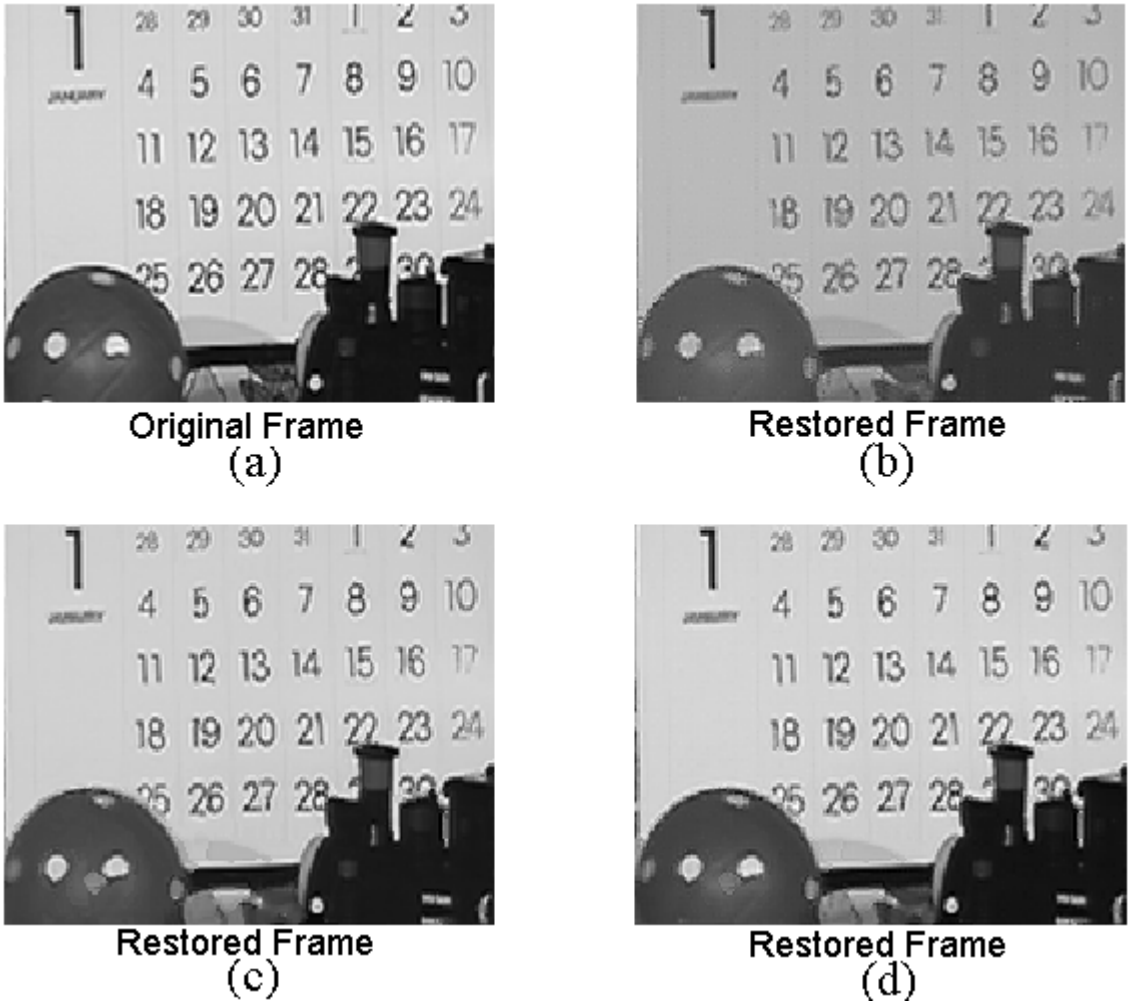

Figure 3.7: Restored frames using different regularization parameters for "Mobile" sequence. (a) original image, (b) super-resolved by $\gamma = 0.025$, (c) super-resolved by $\gamma = 1.5$,(d) super-resolved by $\gamma = 0.175$

We compared the results of the proposed method to methods presented in [F.Brandi (2008)], and [B. Song (2011)]. These algorithms use HR still images for super-resolution of the LR video sequence in different ways, and are explained briefly in the beginning of this chapter. In addition, we compared result of our regularization function to the results of using BTV regularization function in the same way it is used in [Protter *et al.* (2009)]. In this test, we assume one auxiliary HR still image is available every $T = 30$ LR frame, and the 15^{th} frame is going to be super-resolved. A longer time interval compared to previous experiment has been chosen to examine the effectiveness of algorithms when correlation of HR still images and current frame is not very high. Closest HR still images to the frame being super-resolved are 1^{st} and 31^{st} frames. Results for PSNR of the reconstructed frame using different algorithms is represented in Table 1 and indicate better performance of our proposed method. Figure 3.9 and Figure 3.10 illustrate visual quality of restored frames reconstructed using different method. Sample restored frames for "Mobile" sequence show that edges of restored frames using our method are reconstructed with a reasonably high quality compared to other algorithms. It is expected that our method should outperform the conventional regularization function such as BTV because in our method we assumed that we access to additional information in the form of HR still images of the the scene. Therefore comparison to those method is for demonstrating how those additional information are exploited effectively, and does not have any other meaning. Methods suggested by [F.Brandi (2008)], and [B. Song (2011)] exploit the same extra information we used. Therefore, comparison to those method shows that our algorithm can exploit the information of auxiliary HR still images more effectively.

Table 3.1: PSNR comparison for frame reconstruction

	Bicubic	F.Brandi (2008)	BTV	B. Song (2011)	Proposed
City	30.6	31.6	32.7	32.9	**34.0**
Shields	31.9	31.5	32.8	32.7	**34.1**
Container	26.7	28.9	30.6	33.2	**34.2**
News	29.4	30.4	34.3	36.1	**37.2**
Mobile	22.1	23.4	23.7	25.5	**26.0**
Hall Monit	28.6	30.5	33.5	38.0	**38.1**

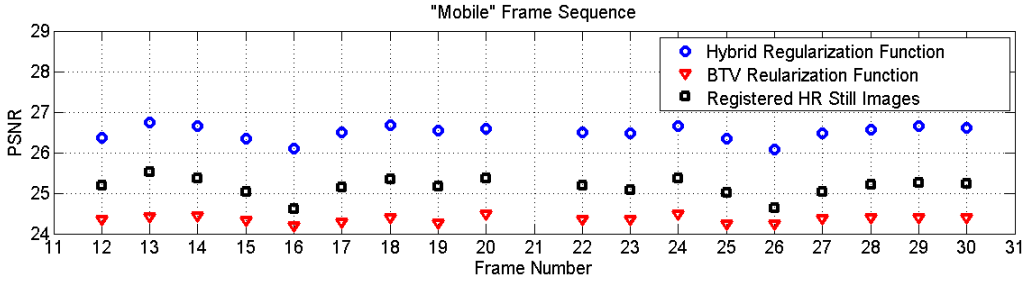

Figure 3.8: Comparison of super-resolution results: " Registration based" vs "Regularized reconstruction approach"

3.7 Conclusion

This chapter presents an approach to super-resolution problem using auxiliary HR still images of the scene. A regularization function is introduced to effectively exploit the information of HR still images as well as the information of the LR frame sequence for super-resolution. Registration of LR frames and HR still images are performed in a fuzzy manner, and many pixels contribute to reconstruct a pixel with different weights to up-sample each frame. Then, de-blurring task is separately performed after up-sampling. We proposed a regularization function for this step which is based on the HR still images. GCV method employed for defining regularization function. A closed form GCV cost function derived which optimum regularization function minimizes it.

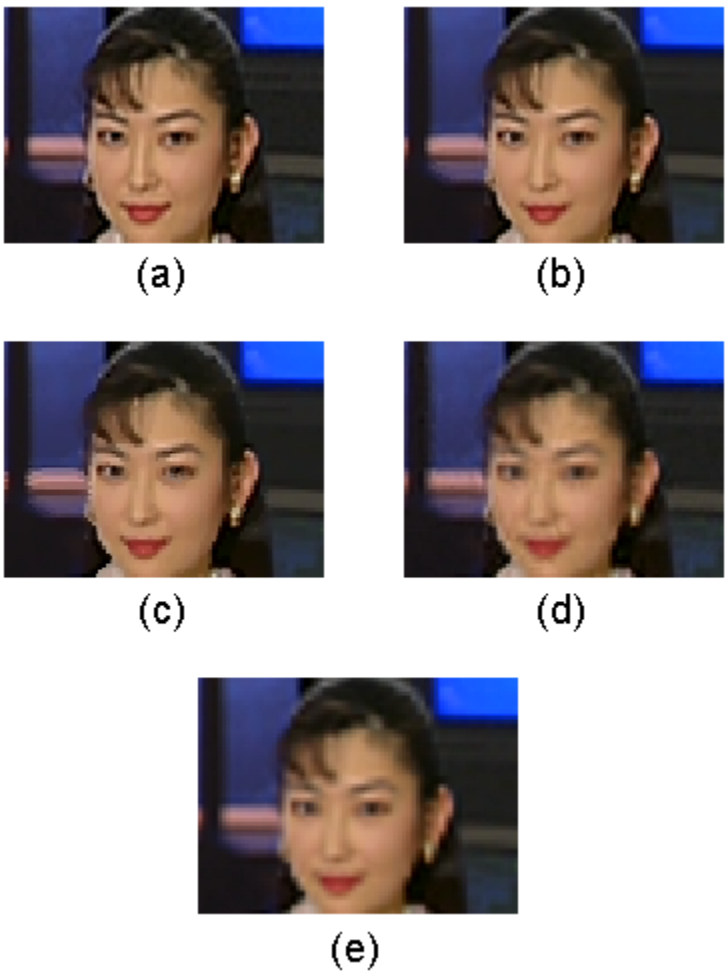

Figure 3.9: Reconstruction of "News" sequence, frame number 15 (a) original, (b) reconstructed using proposed algorithm, (c) BTV regularizer, (d)reconstructed using method proposed in [F.Brandi (2008)], (e) interpolated using bi-cubic method

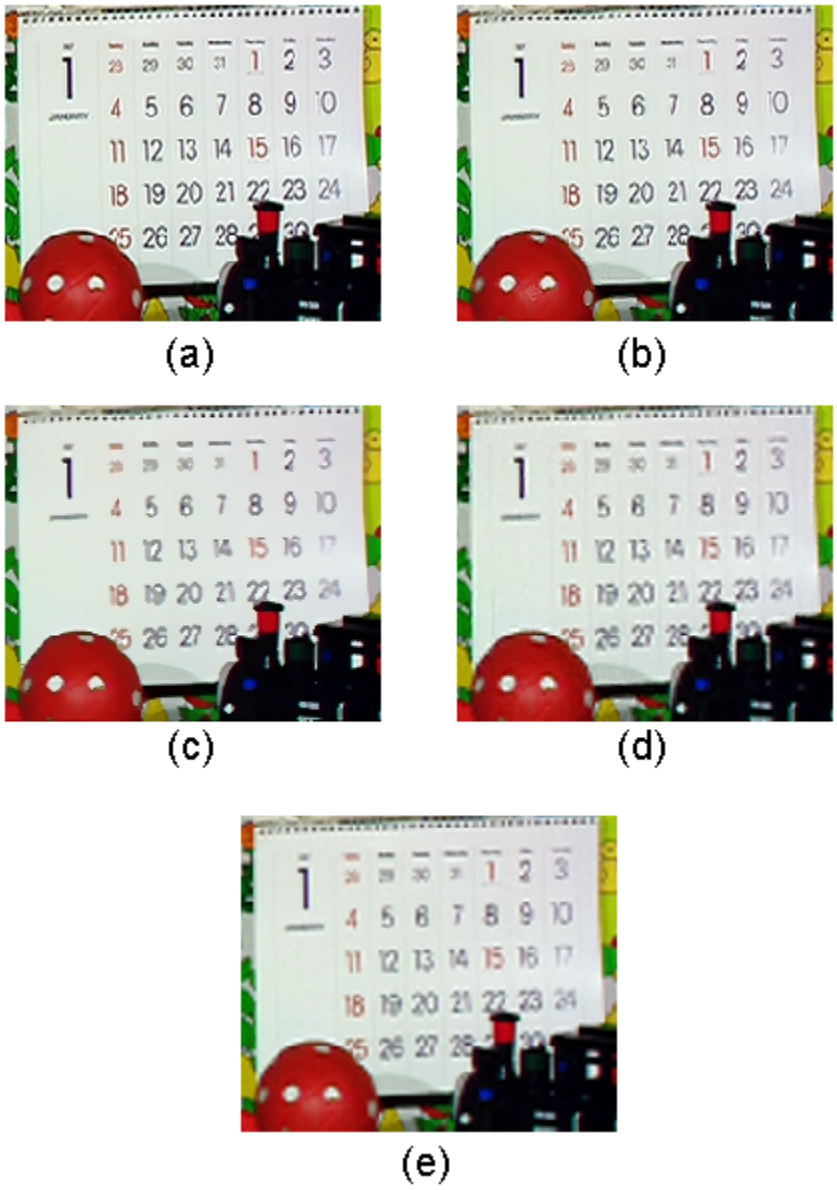

Figure 3.10: Reconstruction of "Mobile" sequence frame number 15. (a) Original, (b) reconstructed using proposed algorithm, (c) BTV regularizer, (d) reconstructed using method proposed in [F.Brandi (2008)], (e) interpolated using bi-cubic method

Chapter 4

Video super-resolution using Multi-view frame sequences

4.1 Introduction

In previous chapter, video super-resolution with the help of high resolution still images was discussed. We assumed these still images are captured at a constant time interval from the same view point that LR video was captured and, and as result, may be considered as a frame sequence with lower temporal but higher spatial resolution. In this chapter, exploiting LR video sequences shot from different view-points is investigated to define how they may carry complementary information to super-resolve the LR sequence at hand. Therefore, different from previous chapter, we do not have access to any HR video frames and we have multiple cameras which are recording the scene with the same resolution from different view-points. Video achieved by multiple cameras from the scene is called multi-view video and has various applications especially for 3D and free view-point video display. We will investigate the use of

simple and common one-dimensional multiple camera setup for multi-view sequence capturing. In this setup each camera is aligned with others on a straight line at a small distance (such as 5 cm). Super-resolution in this chapter deals with the sequences captured by such a setup. Therefore, disparity estimation between frames of different views is simplified, and we can avoid the problem of fast and accurate disparity estimation for general multi-view sequences arises in the field of multi-view video compression. This way, we focus on our special problem which is multi-view video super-resolution. Compared to conventional multi-frame super-resolution scenarios where adjacency between frames is defined along time axis, here we are adding another dimension of adjacency which is the view direction.

4.2 Multi-view Video and Disparity Estimation

Multi-view video is a set of sequences shot from the same scene simultaneously and expands the traditional single-view video sequences in a new direction named "view direction". Multi-view video contains much more redundancy compared to single-view video sequences as sequences from different view-points are capturing data from the same scene. Redundancy is an important notion in the context of signal and video compression, and any sort of redundancy have to be recognized, located, and eliminated in an ideal video compression algorithm. Similarly, existence of redundancy is very essential for resolution enhancement in the context of multi-frame super-resolutions because it can help finding and reconstructing removed and degraded data. For this reason locating and extracting the redundancies in video sequences is also very important to resolution enhancement algorithm designers.

Binocular disparity refers to the difference in image location of an object seen by

the left and right eyes, resulting from the eyes' horizontal separation. Disparity refers to the difference in location of any point of the scene when it is seen from different view points. Depth of objects has a significant effect on their disparity and closer objects have larger disparity than farther ones. Therefore, disparity of an object may be large or small based on its distance. Consequently, common approaches to motion estimation are not effective for disparity estimation as we will need very large search region. Moreover, as disparity refers to finding displacement of objects in frames shot at the same time by different cameras, there is no motion. Therefore, direction of the disparity vector of different objects is only defined by the camera setup and the relative location of cameras. Fortunately, location of the cameras are fixed during the recording and we can say direction of disparity vectors will not change from one time instance to the other [Zhu *et al.* (2010)]. For example, in the test sequences we use in this chapter, eight cameras are used and they are aligned horizontally at a fixed distance from each other. Therefore, direction of disparity of each two frames from two views can be easily determined by finding the direction of lines connecting two cameras to each other in the space. As a result, when one camera is placed 5 cm from the other to its left side, we can say all disparity vectors are almost horizontal vectors pointing to left. In addition, depth of objects does not vary very much, and as a result disparity vectors of the adjacent points of the scene are highly correlated [Zhu *et al.* (2010)]. These facts can be used to design a fast and effective disparity estimator for general multi-view sequences.

In our problem we simply estimate disparity by searching the best match for each block of current frame along a narrow search region elongated in the expected direction of disparity vectors inside the reference frame. Therefore when cameras

are aligned horizontally, the expected disparity direction is horizontal to the left or right, based on the relative location of cameras shooting the current and reference frame. Once disparity between frames of different views is compensated, searching for similarities between them is carried out by the same algorithm we used in the previous chapter (the algorithm applied to adjacent frames in temporal direction).

We will discuss how much information we can extract from other views in a super-resolution application compared to information exist in adjacent frames in the next section.

4.3 Frame Reconstruction using Multi-view Sequences

4.3.1 Multi-view Video Super-resolution

In this section we will discuss how one frame in a multi-view sequence may be super-resolved using temporal neighbors and frames in other views carrying relevant information. The basics for super-resolution using multi-view sequences are the same as what described in the third chapter. The problem consists of up-sampling and de-blurring. The difference is that extra information we access to in the scenario of this chapter is not auxiliary still images and is LR video sequences shot from view-points different from view-point of the main sequence. We used auxiliary still images for de-blurring and for de-noising in the previous chapter, but in this chapter we will do this task by means of conventional regularization functions. In the third chapter we saw how adjacent frames may be used for up-sampling of the current frame. We tried to estimate samples removed from the HR grid of the image sensor by searching for them in the adjacent frames. We take advantage of frames of other views in

multi-view video super-resolution by incorporating them in this search process. The main difference between frames of other views and temporal adjacent frames is that frames from other views do not contain motions and are associated through disparity. This difference enables them to add information for super-resolution especially when occlusion happens in temporally adjacent frames due to motion. Therefore, we should consider collocated frames of the other views in parallel to the temporal adjacent frames in the up-sampling process after disparity compensation. As we said in the last chapter, diamond search is a powerful algorithm to estimate motion between temporally adjacent frames. This algorithm is not capable of tracing long disparity vectors to help us find similar pixels in frames of different views, and is suitable for finding local short displacements. As a result, we should compensate disparity between two frames of different views first, and then apply diamond search to find similarities between them. Disparity estimation we apply here does not have to be very precise as it is followed by diamond search, and is only required to compensate for the long disparity displacement between frames of different views. Up-sampling process is similar to what explained in previous chapter. All pixels of each LR frame are mapped properly on the HR grid. Then missing pixels on the HR grid are primarily interpolated. Each pixel of the interpolated frame (including original and interpolated pixels) is reconstructed by finding all candidate pixels in adjacent frames, weighting them followed by averaging them. These candidate pixels are chosen from original pixels not interpolated pixels. Adjacent frames are chosen both in temporal and view direction for the multi-view super-resolution. The weight for each contributing pixel from a temporally adjacent frame is expressed in previous chapter. Similarly defined as (3.3), the weight for a contributing pixel located at (i, j) from a frame of other

view is as follows:

$$w(k,l,i,j,m) = \exp(-\left\|\mathbf{R}_{(k,l)}\boldsymbol{y} - \mathbf{R}_{(i,j)}\hat{\boldsymbol{y}}_m\right\|_1^1 / 2\delta^2 S^2) \tag{4.1}$$

where again, $\mathbf{R}_{(m,n)}$ is the operator extracting the patch centered at the location (m,n). δ controls the relative contribution of different pixels based on their local similarity to the pixel located at (k,l), and S^2 is the patch size. $\hat{\boldsymbol{y}}_m$ is the disparity compensated version of LR frame captured at time t and is from the m^{th} view. There is no difference between this formula and the formula for weighting pixels of temporally adjacent frames apart from the fact that we perform disparity estimation between two frames before measuring the similarities. After this, all candidate pixels are weighted and we average them according to their weight to reconstruct the pixel located at (k,l) inside the current interpolated frame as follows:

$$\boldsymbol{z}(k,l) = \frac{\sum_{t=1}^{T} \sum_{(i,j)\in\Omega^t{}_{(k,l)}} w(k,l,i,j,t)\boldsymbol{y}_t(i,j) + \sum_{m=1}^{M} \sum_{(i,j)\in\Omega^m{}_{(k,l)}} w(k,l,i,j,m)\boldsymbol{y}_m(i,j)}{\sum_{t=1}^{T} \sum_{(i,j)\in\Omega^t{}_{(k,l)}} w(k,l,i,j,t) + \sum_{m=1}^{M} \sum_{(i,j)\in\Omega^m{}_{(k,l)}} w(k,l,i,j,m)} \tag{4.2}$$

where T is the number of temporally adjacent LR frames used for up-sampling process, $\Omega^t{}_{(k,l)}$ is the set of all candidate pixels inside the search region for the pixel located at (k,l) in the adjacent LR frame $\boldsymbol{y}_t$. M is the number of views which is used in the up-sampling process, and $\Omega^m{}_{(k,l)}$ is the set of all candidate pixels inside the search region for the pixel located at (k,l) in the LR frame of view m shown by $\hat{\boldsymbol{y}}_m$ in the above formula.

The goal of this section is to study the performance of super-resolution algorithm when multi-view video is applied as input and compare it to the case of mono-view

sequence. The above algorithm explained the fusion step for combining frames of a multi-view video sequence to achieve an up-sampled frame. The fusion algorithm for mono-view video explained in the previous chapter and is very similar to fusion algorithm explained for multi-view in this chapter. To compare the result of multi-view and mono-view fusion algorithms, we should eliminate the effect of PSF of the image sensor's LR grid and the noise from $\boldsymbol{z}$. The choice of regularization function we used in this chapter does not matter as we want to compare the final performance of mono-view and multi-view fusion algorithms. The regularization function we use simply smoothes the restored frame in horizontal and vertical directions, and is given by:

$$\Upsilon(\boldsymbol{x}) = \sum_{i=0}^{1} \sum_{\substack{j=-1 \\ j\neq 0}}^{1} \left\| \boldsymbol{x} - \boldsymbol{S}_x^i \boldsymbol{S}_y^j \boldsymbol{x} \right\|_2^2 \tag{4.3}$$

where $\mathbf{S}_x^m$ and $\mathbf{S}_y^m$ are two matrix operators, and shift the image by m pixel in horizontal and vertical directions, respectively. This regularization function is the generalized form of the simpler regularization function which only takes the derivative in the horizontal or vertical direction [M. Elad (2007)]. Putting this regularization function into the minimization problem we obtain:

$$\begin{aligned} e(\boldsymbol{x}) &= \|\boldsymbol{H}\boldsymbol{x} - \boldsymbol{z}\|_2^2 + \gamma \Upsilon(\boldsymbol{x}) \\ \hat{\boldsymbol{x}} &= \min_x \{e(\boldsymbol{x})\} \end{aligned} \tag{4.4}$$

We proceed by taking derivative of the cost function with respect to $\boldsymbol{x}$ and equating it to zero:

$$\frac{de(\boldsymbol{x})}{d\boldsymbol{x}} = 2\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{x} - 2\boldsymbol{H}^T\boldsymbol{z} + \gamma \frac{d}{d\boldsymbol{x}} \sum_{i=-1}^{1} \sum_{\substack{j=-1 \\ i+j\geq 0}}^{1} ((\boldsymbol{I} - \boldsymbol{S}_x^i \boldsymbol{S}_y^j)\boldsymbol{x})^T ((\boldsymbol{I} - \boldsymbol{S}_x^i \boldsymbol{S}_y^j)\boldsymbol{x})$$

$$= 2\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{x} - 2\boldsymbol{H}^T\boldsymbol{z} + 2\gamma \sum_{i=-1}^{1} \sum_{\substack{j=-1 \\ i+j\geq 0}}^{1} (\boldsymbol{I} - \boldsymbol{S}_x^i\boldsymbol{S}_y^j)^T(\boldsymbol{I} - \boldsymbol{S}_x^i\boldsymbol{S}_y^j)\boldsymbol{x}$$

$$= 2\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{x} - 2\boldsymbol{H}^T\boldsymbol{z} + 2\gamma \sum_{i=-1}^{1} \sum_{\substack{j=-1 \\ i+j\geq 0}}^{1} [\boldsymbol{I} - \boldsymbol{S}_x^i\boldsymbol{S}_y^j - (\boldsymbol{S}_x^i\boldsymbol{S}_y^j)^T + (\boldsymbol{S}_x^i\boldsymbol{S}_y^j)^T(\boldsymbol{S}_x^i\boldsymbol{S}_y^j)]\boldsymbol{x}$$

$$= 2\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{x} - 2\boldsymbol{H}^T\boldsymbol{z} + 2\gamma \sum_{i=-1}^{1} \sum_{\substack{j=-1 \\ i+j\geq 0}}^{1} [2\boldsymbol{I} - \boldsymbol{S}_x^i\boldsymbol{S}_y^j - \boldsymbol{S}_y^{-j}\boldsymbol{S}_x^{-i}]\boldsymbol{x}$$

and the solution for $\boldsymbol{x}$ is:

$$\begin{aligned} \hat{\boldsymbol{x}} &= (\boldsymbol{H}^T\boldsymbol{H} + \gamma\boldsymbol{\Lambda})^{-1}(\boldsymbol{H}^T\boldsymbol{z}) \\ \boldsymbol{\Lambda} &= \sum_{i=0}^{1} \sum_{\substack{j=-1 \\ i+j\geq 0}}^{1} (2\boldsymbol{I} - \boldsymbol{S}_x^i\boldsymbol{S}_y^j - \boldsymbol{S}_y^{-j}\boldsymbol{S}_x^{-i}) \end{aligned} \tag{4.5}$$

γ is the regularization parameter and can be easily determined in the same way we determined in the previous chapter by GCV method. Employing GCV method the optimum γ is that minimizes the following expression:

$$V(\gamma) = \frac{\frac{1}{S}\left\|(\boldsymbol{A} - \boldsymbol{I})\boldsymbol{z}\right\|^2}{[\frac{1}{S}tr(\boldsymbol{I} - \boldsymbol{A})]^2} \tag{4.6}$$

where $\boldsymbol{A}$ is defined as follows:

$$\boldsymbol{A} = \boldsymbol{H}(\boldsymbol{H}^T\boldsymbol{H} + \gamma\Lambda)^{-1}\boldsymbol{H}^T$$

4.3.2 3D Frame Compatible Video Format Reconstruction

3D video is recorded and transmitted in a format which contains left and right view to be displayed on 3D TVs. These two views are effective for representing depth of

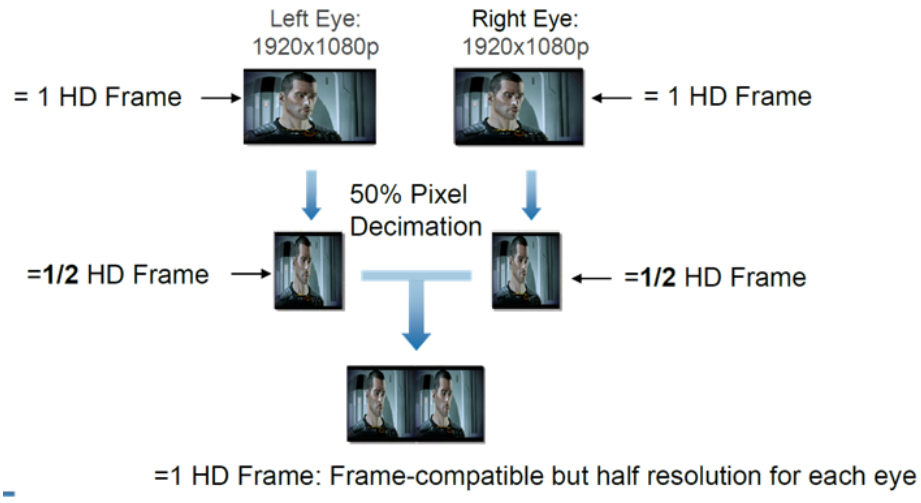

Figure 4.1: Left-right frame compatible stereo video format

the scene. Different formats exist for representing and transmitting 3D video. Frame compatible formats are among the most common formats being used today. Frame compatible formats combine frames of two views at each time instance in order to concatenate and send them in one frame. Before emerging 3D video existing video production and transmission infrastructure and equipments was designed for mono-view video, and frame compatible formats are used widely to facilitate using this infrastructure with minimum changes. Obviously, this format requires transmitter to reduce number of pixels of each frame to half to put right and left view frames in one frame. For example, in side-by-side frame compatible format half of the columns of each frame is removed, then halved frames are concatenated to form a complete frame as shown in Figure 4.1.

At the receiver, removed columns should be interpolated by means of an interpolation methods. Top-bottom 3D format is another way for conveying stereo video in frame compatible mode, where half of the rows of frames are removed and halved left and right frames of each time instance are put one at the top of the other to form one

frame. Removing half of the rows or columns is similar to what is called video interlacing traditionally. It is worth mentioning that the process of transforming stereo video (contains complete left and right view) into side-by-side and top-bottom 3D frame compatible formats is not exactly the same as traditional interlacing approach. But, the act of removing half of rows or columns which occurs in this transformation is called interlacing hereafter in this thesis. De-interlacing refers to algorithms used for interpolating interlaced frame reconstruction. Although many different video de-interlacing method exist we are interested in NLM-based video de-interlacing method presented in [R. Dehghannasiri (2012)] because it is very similar to the frame fusion method we used in this thesis. The NLM-based de-interlacing method introduced in that work, reconstructs each pixel of removed rows or columns by finding and weighting candidate pixels in that frame or temporally adjacent frames and taking average of them. Searching, weighting and reconstruction approach is almost the same as what we did in up-sampling step of super-resolution in previous chapter with minor differences.

In this section we want to extend that de-interlacing method to frame compatible 3D video format reconstruction. As explained, in side-by-side frame compatible format, left and right views at each time instance are interlaced and concatenated to fit into one frame. Receiver is required to interpolate them in some ways to display a full-frame video. For this aim, we use the basics of frame fusion method explained in the previous section. Pixels of removed rows or columns are reconstructed by taking average of all weighted pixels of the adjacent frames and frame of the other view. Therefore, we are extending NLM-based reconstruction approach of [R. Dehghannasiri (2012)] by using frames of the other views. As no noise or blurring is occurred

prior to removing rows or columns, there is no need to perform de-blurring and de-noising step, and after up-sampling frame is reconstructed. We will see how using information of the other views leads to better reconstruction of interlaced frames.

4.4 Multi-view Video Super-resolution by means of Auxiliary High Resolution Still Images

In chapter 3, we presented a super-resolution algorithm for mono-view frame sequences which depends on a sequence of auxiliary HR still images taken in parallel to the LR video frame sequence. We assumed that the camera can deliver a sequence of HR still images with a time period much larger than the time period of the LR frame sequence. Finally, we combined information of these two sequences to produce a HR frame sequence. For that algorithm, we assume the super-resolution algorithm is performed on the sequences before encoding and video encoder encodes the super-resolved HR frame sequence. Therefore, compression does not affect the performance of the algorithm.

In this section we extend the idea of the third chapter and use it for multi-view video. We saw in this chapter that exploiting more than one view can enhance frame up-sampling and consequently enhance the super-resolution process. Therefore, we expect to observe a performance improvement in super-resolution by using a sequence of auxiliary HR still images for multi-view sequences.

Efficient multi-view coding for 3D and free view-point video display requires elaborated coding methods to exploit huge temporal and inter-view redundancies. This is usually done by a central system that is fed by videos coming from different views,

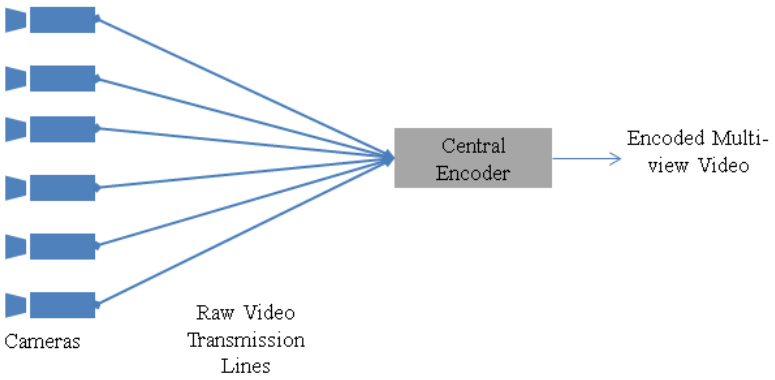

Figure 4.2: Central encoding structure for multi-view video coding used in H.264/MVC standard

and is responsible for compressing them jointly by performing motion and disparity estimation (See Figure 4.2). International standard for multi-view video coding (MVC) that is an extension to H.264/AVC uses the same structure. These encoding methods, burdens a huge complexity on the encoder and are in need of broadband data line for delivering raw data from cameras to the central unit, which limits their applications. Most of this complexity is due to the motion and disparity estimation process at encoder where efforts have been taken to reduce it or relocate it to the decoder [L. Shen (2001)] ,[X. Guo (2008)]. Since disparity and motion estimation should be performed for each pixel of all frames in the sequence, the complexity of these tasks and consequently encoder increases as the resolution of frames increases. The multi-view video super-resolution algorithm we explain in this chapter could be an extra computation burden on the encoder side if we apply it before the encoding as shown in Figure 4.3. This problem has two sides. First, encoder has to encode the super-resolved HR sequences. Second, super-resolution algorithm computation is added to the encoder side. The advantage of this structure is that lossy compression can not affect the performance of super-resolution algorithm because super-resolution

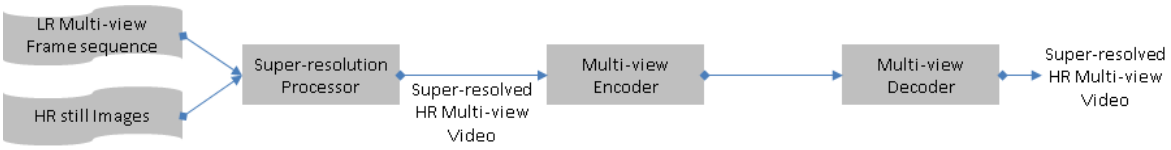

Figure 4.3: Super-resolution at encoder

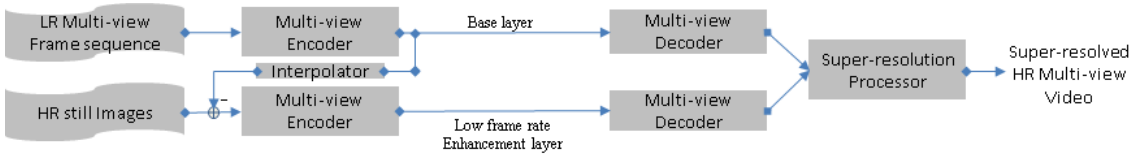

Figure 4.4: Super-resolution at decoder

is performed on the sequences before any compression. In another option, we can encode the LR layer and the HR layer separately by the encoder and super-resolution is performed at the decoder side, after LR and HR layers are decoded as shown in Figure 4.4. In this case, encoder does not have to encode the previously super-resolved multi-view sequence which is high-resolution. Considering HR still images are taken at a constant period along the time in different views, they actually form a low rate HR multi-view sequence which should be encoded by the encoder (See Figure 4.5). The other sequences is the LR frame sequence, which should be encoded in parallel. The HR multi-view sequence of still images does not involve a large amount of computation due to its low rate. Therefore, computation burden of the super-resolution process is shifted to the decoder side, which helps maintaining a balance between encoder and decoder complexity. Super-resolution algorithm is performed on the compressed sequences in this option. To employ this method, the LR multi-view sequence is encoded by the means of H.264/MVC at the regular frame rate as the base layer. In addition, sequence of HR still images is encoded at its low frame rate as the enhancement layer. An illustration of LR and HR sequences in this scheme is

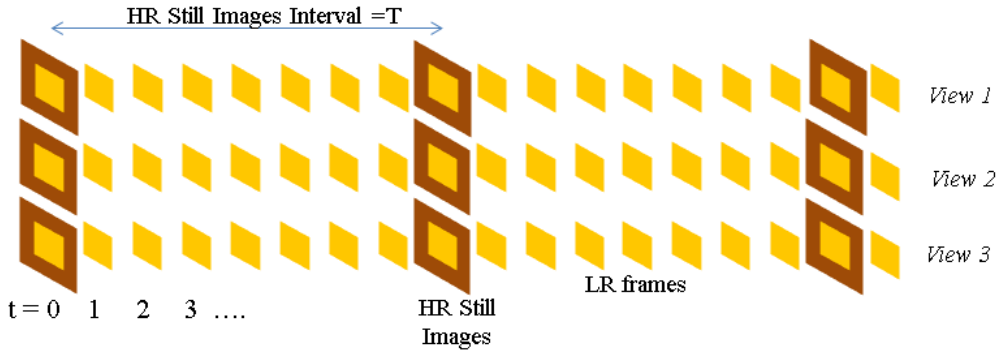

Figure 4.5: HR and LR sequences in the proposed scheme

given by Figure 4.5.

Encoding LR multi-view sequence reduces encoder complexity, and mitigates the need of a broadband inter-camera channel for transmitting raw HR video sequences to the central encoder. In addition, we recover HR multi-view sequence at the decoder by employing the enhancement layer of HR still images in a super-resolution algorithm.

In this section, a super-resolution algorithm for the frame reconstruction at the decoder is explained. In fact, video enhancement algorithm in this section is purely a combination of the proposed algorithms discussed in previous sections.

4.4.1 Encoder Side

LR sequences are encoded using the H.264/MVC as the base layer. Sequence of HR still images plays the role of enhancement layer. Since HR still images are produced by the camera at a low rate, they are sparsely placed compared to LR frames (*e.g.* with period of $T = 10$ LR frames for each HR still image). The decoder is expected to exploit sparsely placed HR still images to enhance the resolution of all frames of LR sequences. To encode the HR still images we use primarily up-scaled version of corresponding LR frame as its predictor and resulting residue frames are encoded

using H.264/MVC as shown in Figure 4.4. This process may seem unusual as H.264 encoder is designed to use adjacent frames in time and collocated frames in other views as reference for coding each frame. Encoding the original sequence of HR still images is not efficient in terms of compression as it's rate is too low, and there is not correlation between adjacent frames along the time direction. In the current scenario, we have access to LR version of each HR still image, and we can use them as predictors for the sequence of HR still images if we simply interpolate them. Thus, we interpolate each LR frame associated with a HR still image by a simple interpolation method such as bilinear to produce a good primary reference for the corresponding HR still images as inter layer prediction.

4.4.2 Decoder Side

Frame restoration algorithm at the decoder side should exploit LR and HR layers to recover a HR sequence for each view. The proposed method for frame reconstruction at the decoder includes two steps. Both are described in details previously. In the first step, each LR frame is up-sampled using the up-sampling method described for multi-view sequences in this chapter. Then de-blurring is performed to eliminate the effect of camera PSF. As compression has degraded all LR decoded frames, the up-sampled LR frame cannot be related to the ideal HR frame by a simple linear degradation model introduced in previous chapters. As we do not want to model the effect of compression, de-blurring cost function simply consists of the regularization function (*i.e.* regularization parameter is equal to zero). Regularization function used for the de-blurring step is the same regularization function used in the chapter 3 to incorporate auxiliary still images in reconstruction process. Therefore, final

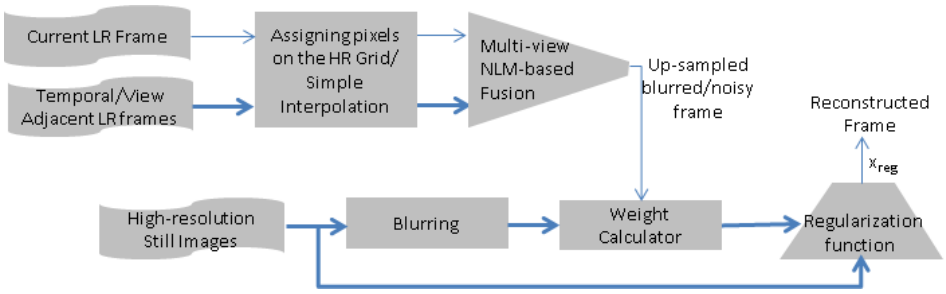

Figure 4.6: Full diagram of the proposed method

restored frame is equal to $\hat{\boldsymbol{x}}_{reg}$ described by (3.16) in the previous chapter. It is worth mentioning, modeling the compression effect makes it possible to incorporate the up-sampled frame in the de-blurring process directly. This can be considered as future work related to this thesis. Currently, the output of up-sampling process is used in de-blurring step to define weights used in the regularization function, and do not directly contribute to the reconstruction of current frame through a degradation model (see Figure 4.6 and compare it to the Figure 3.3).

4.5 Experiments

In the first part of this section we will examine the performance of using multi-view sequence for super-resolution. We compare its result to results of mono-view super-resolution and investigate how much performance improvement we may get from other views for super-resolution. In the second part, we examine 3D frame-compatible video format reconstruction algorithm explained in previous section of this chapter.

All tests are performed by three multi view sequences consists of "Race1", "Ballroom" and "Exit" with resolution of 480×640. Each of these sequences contains eight views shot by one directional camera setup. Race1 shows a match between racing cars

and represents fast object and camera motion. Ballroom shows many dancers dancing in the hall and contains fast motions and occluded areas between frames. Exit is a slow video showing people going out through a door. For evaluating multi-view video super-resolution algorithm, we blurred and down-sampled these three sequences by a factor of 2, and used them as LR videos. Blurring window which plays the role of PSF of camera is a 3×3 Gaussian window with $std = 0.7$, and contaminating noise's variance is equal to 1. We first avoid using adjacent views of the view we want to super-resolve for up-sampling step. It is the same as mono-view video super-resolution. Then, we include different number of collocated frames of other views in the up-sampling step in different experiments. In one experiment, we include one extra view (the second view) in the super-resolution of the first view. In two other experiments, we include three and seven views in the up-sampling step. This way we can see how efficient is to add one, three and seven extra cameras for super-resolution. In all tests, number of all frames (adjacent frames and frames of other views) involved with up-sampling step is kept constant for maintaining the level of computation complexity. In other words, when we super-resolve the first view without using any other view we use ten temporally adjacent frames of that view (usually five previous, five future) for the up-sampling step. When we involve one more frame from the second view, we decrease adjacent frames involved with up-sampling process to eight. For the cases we include three and seven frames of other views we use six and two temporally adjacent closest adjacent frame in the up-sampling step. The result of super-resolving frames of the first view for the frames between 1 and 35 is depicted in Figure 4.7, 4.8, and 4.9. As can be seen from this figures, even one extra view can have a significant effect on the quality of super-resolved frame, especially when we are dealing with a

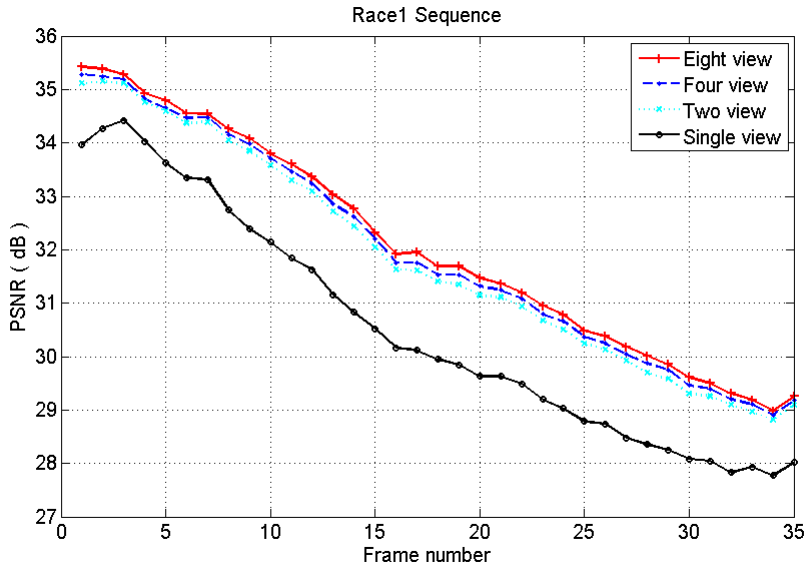

Figure 4.7: Super-resolution of frames of the first view of Race1 multi-view sequence

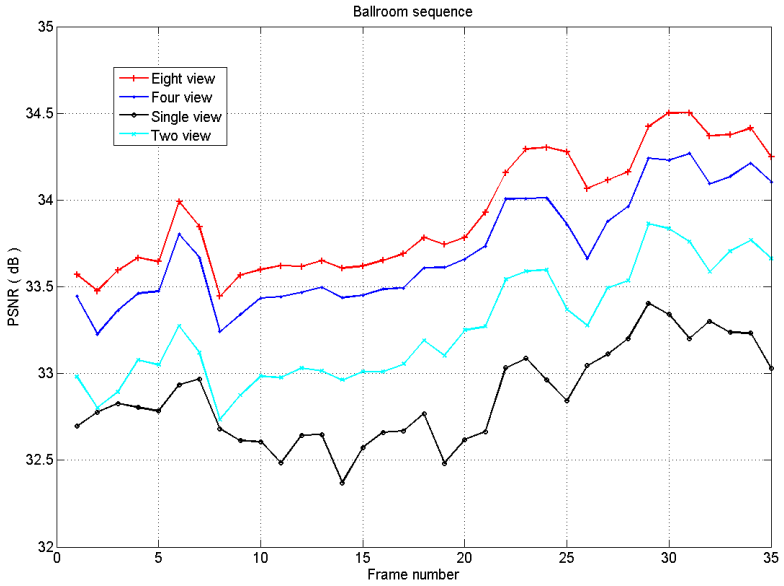

Figure 4.8: Super-resolution of frames of the first view of Ballroom multi-view sequence

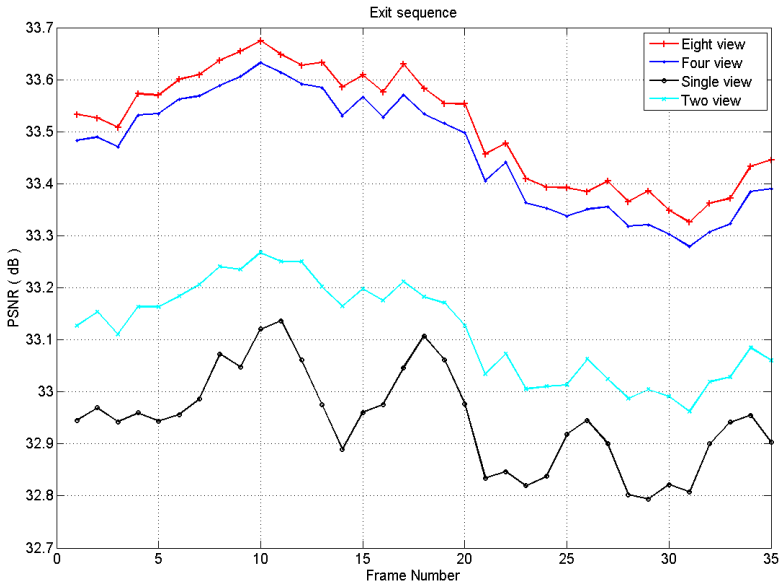

Figure 4.9: Super-resolution of frames of the first view of Exit multi-view sequence

scene which contains fast object and camera motion and one object in the current frame may be occluded in the temporally adjacent frames. Since frames from another view which are shot at the same time instance are not affected by the motion of scene or camera, accessing to them can be very effective for up-sampling, especially when some objects is occluded in the adjacent frames due to motion. This effect is less for videos with slower motion, such as Exit sequence. Accessing more views increases the chance for finding true candidate pixels in frames of those views for reconstruction of missing pixels in the current frame. As can be seen, PSNR of super-resolved frames increases by incorporating more views in the up-sampling process, but rate of this increment is reduced gradually, as correlation between views decreases when views are shot by far cameras.

In the second part of experiments we examined proposed frame compatible 3D

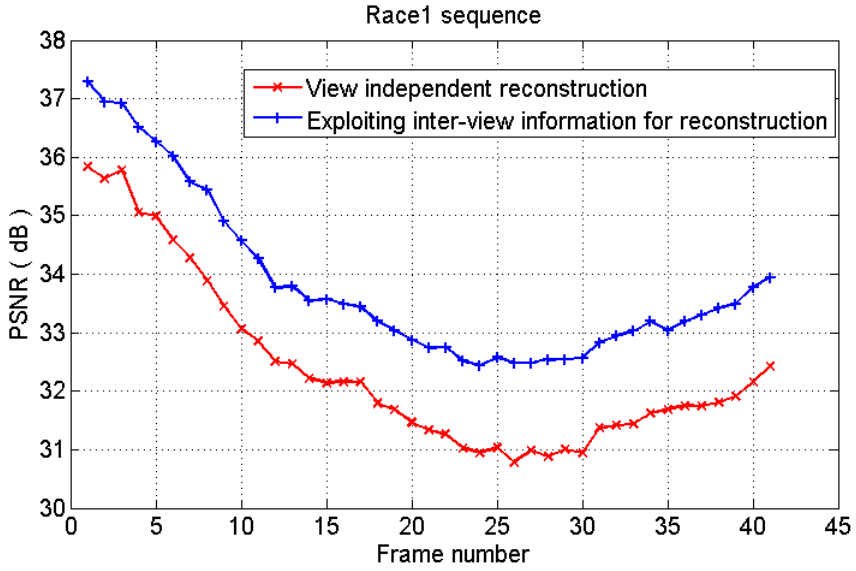

Figure 4.10: Frame compatible 3D video reconstruction for Race1 sequence

video reconstruction algorithm. This algorithm reconstructs removed rows or columns of original frames by employing temporal and view data. We compared results of this algorithm with the results of the algorithm of independent reconstruction of each view. First and second views of "Race1", "Ballroom", and "Exit" sequences are used for test as left and right views. Figure 4.10, 4.11, and 4.12 represent PSNR of reconstructed full frames of the left view using proposed algorithm which uses both view and temporal information. It also depicts PSNR of left view frames reconstructed independently from the right view. It can be inferred from these graphs that incorporating the other view for reconstructing one view can be very effective when camera and scene motion is fast such as Race1 sequence. We achieved 1.3 dB improvement on average when reconstructing left view of this sequence using both temporal and inter-view information. The average amount of improvement is 0.7 dB for Ballroom sequence and 0.23 dB for Exit sequence. This amount and

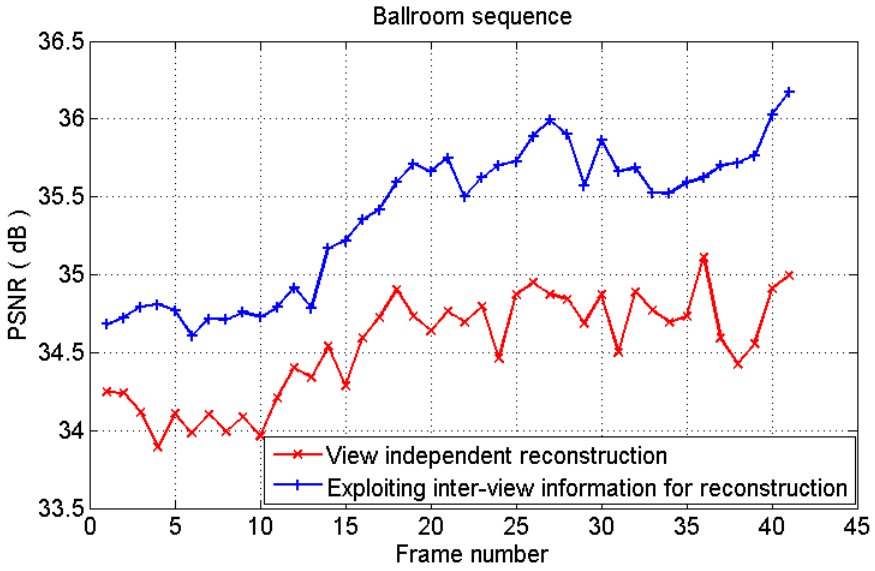

Figure 4.11: Frame compatible 3D video reconstruction for Ballroom sequence

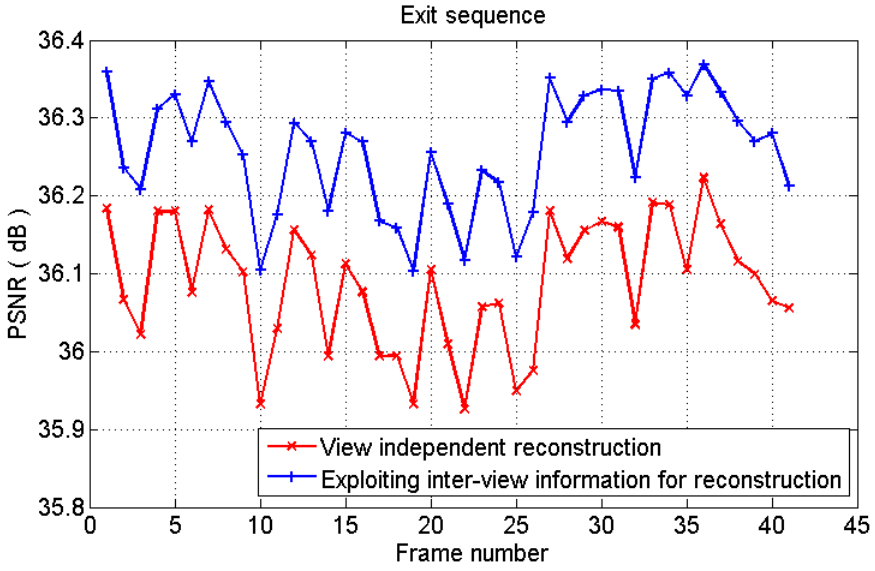

Figure 4.12: Frame compatible 3D video reconstruction for Exit sequence

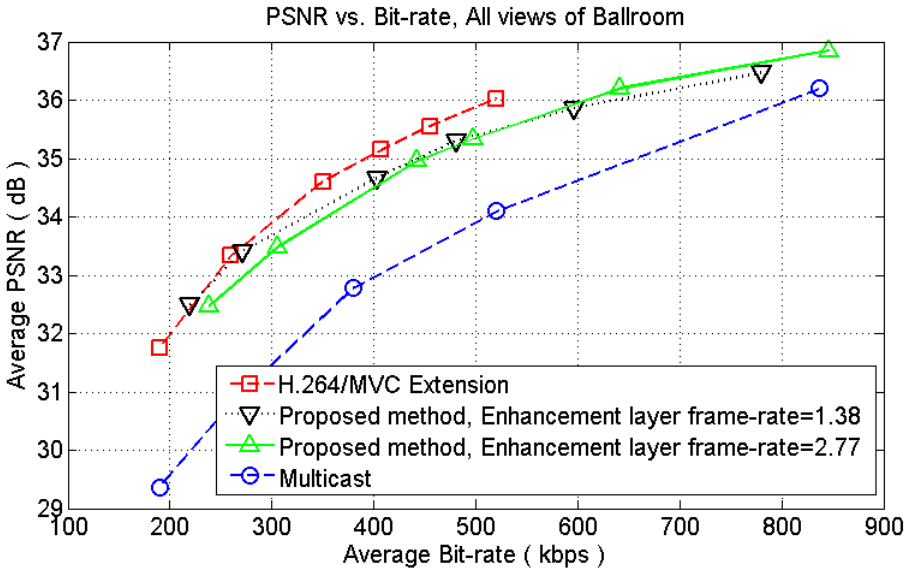

Figure 4.13: Performance of multi-layer MVC scheme compared to regular mono-layer MVC for "Ballroom" sequence

efficiency of proposed algorithm decreases as the video becomes slower in terms of scene and camera motion. Finally we evaluate the multi-layer multi-view coding scheme presented in the previous section. MVC extension of H.264/ AVC version 8.5 is used for encoding LR frames of the base layer, and low rate sequence of HR still images as the enhancement layer separately. We change quantization parameter (QP) of the base layer encoder from 19 to 28 to achieve different bit-rates. Enhancement layer QP is set to be 2 levels higher than the base layer for all tests. Therefore, enhancement layer to base layer bit-rate ratio remains below 20 percent. We examine the performance of the algorithm for two enhancement layer rates in this experiment equal to 1.38 and 2.77 kb/s. These values are associated with T (HR still images interval) equal to 18 and 9 respectively. Figure 4.13, and Figure 4.14 shows average output PSNR of the proposed method for different average bit-rates using Ballroom and Exit sample video sequences. We compare the performance of proposed method

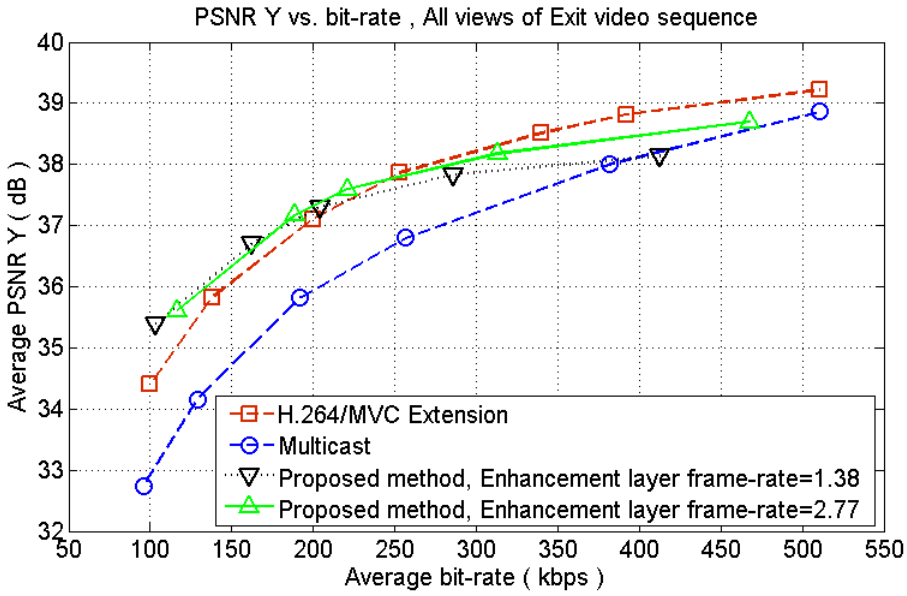

Figure 4.14: Performance of multi-layer MVC scheme compared to regular mono-layer MVC for "Exit" sequence

to the performance of two other coding possibilities. In these scenarios, multi-view video sequence has been shot originally high-resolution. In the first case, we encode HR multi-view frame sequences using MVC extension of H.264. In the second test, each HR view is encoded separately using H.264/AVC (multicast). Experiments show that the performance of proposed method for both sequences is comparable to H.264/MVC and multicast. Moreover, the average PSNR for output of proposed method is higher for bit-rates lower than 250 kbps compared to MVC extension of H.264 for Exit video sequence. Experiments show better results for the "Exit" sequence compared to "Ballroom" sequence as "Exit" sequence contains less fast motions in the scene. As explained in the previous section, cost function used for de-blurring process depends only on the HR still images, and occlusion can reduce the quality of restored de-blurred frame in this way. It can be seen the performance of proposed method decreases for the "Ballroom" sequence which contains fast motions

and occluded areas between frames.

4.6 Conclusion

In this chapter we studied exploitation of inter-view information for super-resolution. Two applications of it including multi-view super-resolution and frame compatible 3D video reconstruction are investigated in this chapter. An algorithm for multi-view sequence super-resolution proposed and simulation results showed that using even two cameras can lead to a reasonable improvement in quality of super-resolved frame. A solution for frame compatible 3D video reconstruction employing inter-view information proposed, and simulation results show that exploiting inter-view information can lead to better output in terms of PSNR quality, especially when video contains relatively fast motion. A multi-layer MVC scheme and associated enhancement algorithm for decoder side proposed in this chapter. Simulation results showed that proposed algorithm can be efficient for MVC in terms of PSNR for low bit-rates. Moreover, the algorithm cannot perform as well for the sequences containing fast motions.

Chapter 5

Conclusion and Future Works

In this thesis a number of algorithms to incorporate extra information in the super-resolution process of a low-resolution video sequence are proposed. In the first chapter, we studied the super-resolution problem using auxiliary HR still images of the scene. A cost function is introduced in that chapter reconstructs each frame of the low-resolution sequence using the available HR still images, and incorporates HR still images in the super-resolution process. Experiment results showed that using auxiliary still images to form the regularization function for the super-resolution inverse problem obviates our need to use conventional regularization functions and leads to better results. We then extended the thesis to super-resolution process for multi-view sequences. We studied how incorporating and fusing frames from more than one view in the super-resolution process can enhance the super-resolution process. We found out that information of other views can be helpful when motion in the scene is very fast. Finally we suggested an approach for multi-layer multi-view video super-resolution. Simulation results showed that the proposed algorithm can be efficient

for MVC in terms of PSNR for low bit-rates. Although, the algorithm cannot perform as well for the sequences containing fast motions, but it is designed to enhance the spatial resolution of decoded multi-view sequence shot by a dual-mode camera. As compression degrades the low-resolution multi-view sequence frames non-linearly after they are degraded by PSF of camera images sensor, we did not incorporate low-resolution decoded frames at the decoder directly to reconstruct the super-resolved frame and used them only for a weight-calculation task. However, compression effect can be modeled and this model can be used to incorporate low-resolution frames in super-resolution process directly in the multi-view video super-resolution case similar to single-view super-resolution case. This problem remains as a future work have may be done later in the direction of this thesis to enhance the performance of the multi-view video super-resolution algorithm using auxiliary still images.

Bibliography

Antoni Buades, Bartomeu Coll, Jean-Michel Morel (2011). Non-local Means Denoising. *Image Processing On Line*.

B. Song, S. J. (2011). Video super-resolution algorithm using bi-directional overlapped block motion compenstion and on-the-fly dictionaty training. *IEEE Transaction on Circuitd and Systems for Video Technology*, **21**(3), 274 – 285.

Chan, T., Osher, S., and Shen, J. (2001). The digital tv filter and nonlinear denoising. *Image Processing, IEEE Transactions on*, **10**(2), 231 –241.

Craven, P. and Wahba, G. (1979). Smoothing noisy data with spline functions. *Numerische Mathematik*, **10**(31), 377–403.

Davis, P. J. (1979). *Circulant Matrices*, chapter "Multipleframe image restoration and registration". Wiley, New York.

Donoho, D. L. and Johnstone, J. M. (1994). Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, **81**(3), 425–455.

Farsiu, S., Robinson, M., Elad, M., and Milanfar, P. (2004). Fast and robust multiframe super resolution. *Image Processing, IEEE Transactions on*, **13**(10), 1327 –1344.

F.Brandi, R. Q. (2008). Super-resolution of video using key frames and motion estimation. In *15th IEEE International Conference on Image Processing, ICIP*, volume 24, pages 1608 – 1611, San Diego, California, USA.

Hansen, P. C. (2000). The l-curve and its use in the numerical treatment of inverse problems. In *in Computational Inverse Problems in Electrocardiology, ed. P. Johnston, Advances in Computational Bioengineering*, pages 119–142. WIT Press.

Huang, H.-Y., Conge, P., and Huang, L.-W. (2011). Cmos image sensor binning circuit for low-light imaging. In *Industrial Electronics and Applications (ISIEA), 2011 IEEE Symposium on*, pages 586 –589.

L. Shen, Z. L. (2001). Low-complexity mode decision for mvc. *IEEE Transactions on Circuits and Systems for Video Technology*, **21**(6), 837 – 843.

Liu, X. and El Gamal, A. (2003). Synthesis of high dynamic range motion blur free image from multiple captures. *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, **50**(4), 530 – 539.

LuminousLandscape (2012). Understanding lens diffraction.

M. Elad, D. D. (2007). Example-based regularization deployed to super-resolution reconstruction of a single image. *The Computer Journal*, **52**(1), 15–30.

Park, S. C., Park, M. K., and Kang, M. G. (2003). Super-resolution image reconstruction: a technical overview. *Signal Processing Magazine, IEEE*, **20**(3), 21 – 36.

PCO-TECH (2012). Binning.

Pelletier, S., Spackman, S., and Cooperstock, J. (2005). High-resolution video synthesis from mixed-resolution video based on the estimate-and-correct method. In *Application of Computer Vision, 2005. WACV/MOTIONS '05 Volume 1. Seventh IEEE Workshops on*, volume 1, pages 172 –177.

Protter, M., Elad, M., Takeda, H., and Milanfar, P. (2009). Generalizing the nonlocal-means to super-resolution reconstruction. *Image Processing, IEEE Transactions on*, **18**(1), 36 –51.

R. Dehghannasiri, S. S. (2012). A de-interlacing method using kernel based nonlocal-means filtering. In *IEEE International Conference on Image Processing*.

Reeves, S. (1991). Blur identification by the method of generalized cross-validation. *IEEE Transactions on Image Processing*, **1**(3), 301 – 311.

S. Najafi, S. S. (2012). Regularization function for video super-reolsution using auxiliary high resolution still images. In 46^{th} *Asilomar Conference on Signal, Systems and Computers*, Pacific Grove, California, USA.

Takeda, H., Milanfar, P., Protter, M., and Elad, M. (2009). Super-resolution without explicit subpixel motion estimation. *Image Processing, IEEE Transactions on*, **18**(9), 1958 –1975.

X. Guo, Y. L. (2008). Wyner ziv-based multiview video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, **18**(6), 713 – 724.

X. Wu, G. Z. (2010). Video super-resolution for dual-mode digital cameras via scene-matched learning. In *Multimedia Signal Processing (MMSP), 2010 IEEE International Workshop on*, pages 438 –442.

Zhou, Z., Pain, B., and Fossum, E. (1997). Frame-transfer cmos active pixel sensor with pixel binning. *Electron Devices, IEEE Transactions on*, **44**(10), 1764 –1768.

Zhu, S. and Ma, K. (2000). A new diamond search algorithm for fast block-matching motion estimation. *Image Processing, IEEE Transactions on*, **9**(2), 287–290.

Zhu, W., Tian, X., Zhou, F., and Chen, Y. (2010). Fast disparity estimation using spatio-temporal correlation of disparity field for multiview video coding. *Consumer Electronics, IEEE Transactions on*, **56**(2), 957 –964.