COMPARISON OF SINGLE AND MULTI FRAME SUPER RESOLUTION RECONSTRUCTION ALGORITHMS WITH ANALYTICAL INTERPOLATION METHODS

by

Hüseyin ÖZDEMİR

B.S., Electrical and Electronic Engineering, Middle East Technical University, 2003

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical and Electronic Engineering
Boğaziçi University
2009

ACKNOWLEDGEMENTS

I would like to express my sincere thankfulness to my supervisor Prof. Bülent Sankur for his assistance during the preparation of this M.S. Thesis.

I would also like to thank to Prof. Lale Akarun and Assistant Prof. Burak Acar for participating in my thesis jury and for their valuable comments on the thesis.

ABSTRACT

COMPARISON OF SINGLE AND MULTI FRAME SUPER RESOLUTION RECONSTRUCTION ALGORITHMS WITH ANALYTICAL INTERPOLATION METHODS

Generating high resolution image (or image sequence) from low resolution image (or image sequence) has various applications such as image expansion, printing and conversion between different resolution formats. There is a huge amount of study on HR image reconstruction problem in the literature. These methods can be broadly divided into two main classes: analytical reconstruction techniques and super resolution reconstruction techniques. In the former case, reconstruction is established using an interpolation kernel. The original LR image is convolved with interpolation kernel to obtain the continuous data. Then continuous signal is sampled again according to the desired resolution. On the other hand, in SR reconstruction, the idea is to fuse different samples obtained at different time instants from the same object by a single camera. SR reconstruction can be posed in another way. HR image can be reconstructed by combining different samples obtained at the same time instant from the same object by multiple cameras. There should be sub-pixel shifts between sampling locations to make the reconstruction possible.

In this M.S. Thesis, subjective and objective comparison of 5 different analytical interpolation methods and 2 super resolution image reconstruction methods is given. An adaptive filtering approach least mean squares (LMS) filtering and robust super resolution are used for super resolution image reconstruction. SR methods are compared with bicubic interpolation, wavelet based interpolation, edge adaptive interpolation, interpolation using wide sense Markov random fields and interpolation using exponential based kernels. All seven methods are tested on different videos and frames. PSNR and SSIM measurements are given. Also, subjective tests are conducted on the experimental results.

ÖZET

TEK VE ÇOK ÇERÇEVELİ SÜPER ÇÖZÜNÜRLÜK YÖNTEMLERİYLE ANALİTİK ARADEĞERLEME YÖNTEMLERİNİN KARŞILAŞTIRILMASI

Düşük çözünürlüklü imge (veya imge dizisi) kullanarak yüksek çözünürlüklü imge (veya imge dizisi) elde etmenin imge büyütme, baskı ve değişik çözünürlük biçimleri arasında döünüşüm yapmak gibi çeşitli uygulama alanları bulunmaktadır. Literatürde konuyla ilgili birçok çalışma mevcuttur. Bu metodlar kabaca iki ana başlık altında incelenebilir: analitik inşa teknikleri ve süper çözünürlük inşa teknikleri. İlk kısımdaki yöntemlerde, inşa etme işi bir aradeğerleme çekirdeği kullanılarak yapılır. Orjinal düşük çözünürlüklü imge ile aradeğerleme çekirdeğinin evrişimi hesaplanarak sürekli imge sinyali elde edilir. Daha sonra bu sürekli sinyal istenen yüksek çözünürlüğe göre tekrar örneklenerek yüksek çözünürlüklü imge elde edilir. Öte yandan, süper çözünürlüklü inşa yönteminde ise ana fikir, aynı nesneden farklı anlarda aynı kamera tarafından alınan örneklerin birleştirilmesi sonucu yüksek çözünürlüklü imgenin elde edilmesidir. Başka bir şekilde düşünecek olursak, süper çözünürlüklü inşa yöntemi, aynı nesneden aynı anda birden fazla kamera tarafından alınan örneklerin birleştirilmesi sonucu yüksek çözünürlüklü imgenin elde edilmesidir. İnşa işleminin mümkün olabilmesi için farklı örnekler arasında piksel altı kaymalar olmak zorundadır.

Bu yüksek lisans tezinde, 5 farklı analitik ara değerleme yöntemiyle 2 süper çözünürlük inşa yönteminin nesnel ve öznel karşılaştırmaları yapılmıştır. Süper çözünürlük inşa yöntemi olarak bir çeşit uyarlamalı süzgeçleme yaklaşımı olan en küçük ortalama kareler kestirimi yöntemi ve gürbüz süper çözünürlük yöntemleri kullanılmıştır. Bu yöntemlerin sonuçları, 2 boyutlu kübik aradeğerleme, dalgacık tabanlı aradeğerleme, ayrıt uyarlamalı aradeğerleme, Markov rastgele alanları kullanılarak yapılan aradeğerleme ve üstel tabanlı aradeğerleme sonuçlarıyla karşılaştırılmıştır. Bütün yöntemler farklı video ve

imgeler üzerinde test edilmiştir. Sonuçların PSNR ve SSIM ölçümleri yapılmıştır. Ayrıca, sonuçlar üzerinde öznel testler yapılmış ve sonuçları değerlendirilmiştir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii

ABSTRACT iv

ÖZET v

LIST OF FIGURES ix

LIST OF TABLES xii

LIST OF SYMBOLS / ABBREVIATIONS xiii

1. INTRODUCTION 1

1.1. Single Frame SR Image Reconstruction 2

1.2. Multi Frame SR Image Reconstruction 3

1.3. Outline of the Thesis 4

2. SUPER RESOLUTION RECONSTRUCTION METHODS 5

2.1. Interpolation of Non-Uniformly Spaced Samples 5

2.2. Simulate and Correct Methods 12

2.3. MAP and ML Reconstruction Methods 15

2.4. Projection onto Convex Sets 22

2.5. Bounding Ellipsoid Method 24

2.6. Tikhonov Arsenin Regularization 25

2.7. Robust Super Resolution Reconstruction 27

2.8. Adaptive Filtering (LMS) Method 28

2.9. Reconstruction via Alias Removal 29

2.10. Recursive Least Squares 32

3. IMAGE INTERPOLATION METHODS 35

3.1. Cubic Convolution 35

3.2. Cubic B-Spline Interpolation 37

3.3. Lanczos Interpolation 37

3.4. Wavelet Based Image Interpolation 38

3.5. Edge Adaptive Image Interpolation 41

3.6. Image Interpolation Using Wide Sense Markov Random Fields 45

3.7. Exponential Based Interpolation Functions 47

4. IMAGE AND VIDEO QUALITY ASSESMENT METHODS 49

4.1. Peak Signal to Noise Ratio (PSNR) 49
4.2. Universal Image Quality Index 49
4.3. Single Scale Structural Similarity 50
4.4. Multi Scale Structural Similarity 52
4.5. Structural Similarity for Video 52
5. EXPERIMENTAL RESULTS AND DISCUSSION 55
5.1. Objective Quality Assessment 55
5.2. Experimental Setup 66
5.3. Results 68
5.3.1. Outlier Subjects 68
5.3.2. Computation of Subjective Scores 70
5.3.3. Comparison of Subjective Video and Subjective Frame Still Scores 73
5.3.4. Comparison of Subjective and Objective Scores 75
6. CONCLUSION 79
REFERENCES 80
REFERENCES NOT CITED 88

LIST OF FIGURES

Figure 2.1.	Super resolution stages	5
Figure 2.2.	Subpixel shifts	6
Figure 2.3.	Pixel apertures	7
Figure 2.4.	Different Pixel Apertures	9
Figure 2.5.	ROI covered with quadrilaterals	10
Figure 2.6.	Deformation of quadrilaterals	10
Figure 2.7.	Quadrilateral patches and deformation	11
Figure 2.8.	Nonuniform interpolation	11
Figure 2.9.	Block diagram for MAP and ML estimation of HR image	15
Figure 2.10.	Huber function with threshold T	18
Figure 2.11.	Block diagram of the system	21
Figure 2.12.	Aliased spectrum as a summation of unaliased spectra	30
Figure 2.13.	Block diagram of HR reconstruction system	33
Figure 3.1.	Interpolating an HR image from an LR image	35
Figure 3.2.	Different interpolation kernels (a) Nearest neighbor (b) Linear interpolation (c) Cubic B-spline (d) Cubic convolution (eq. 3.2)	36
Figure 3.3.	Amplitude spectra of interpolation kernels, red for nearest neighbor, blue for cubic and green for linear interpolation kernels	36
Figure 3.4.	B-spline functions	37

Figure 3.5. (a) *sinc(x)*, (b) *sinc(x/2)*, (c) *sinc(x)sinc(x/2)* 38

Figure 3.6. Original image and its sub-bands 39

Figure 3.7. Image interpolation in wavelet domain 40

Figure 3.8. Edge with k<1 43

Figure 3.9. One possible position for the edge to be reconstructed: a, b, c, d are LR pixels and m is HR pixel 43

Figure 3.10. Two dimensional case in edge adaptive interpolation 44

Figure 3.11. Computation of deterministic and indeterministic parts of HR image . . 45

Figure 3.12. The interpolation coordinate (yellow one) and neighbor pixels (red ones) 48

Figure 4.1. Computation of video SSIM value 54

Figure 5.1. Implemented SR algorithms and interpolation methods 55

Figure 5.2. PSNR values of each frame in sequences 1, 2 and 3 for all methods 59

Figure 5.3. SSIM values of each frame in sequences 1, 2 and 3 for all methods 60

Figure 5.4. 100th frames of the 1st, 2nd and 3rd video sequences, respectively 61

Figure 5.5. Error images of all methods for 100th frame of 1st video 63

Figure 5.6. Error images of all methods for 100th frame of 2nd video 64

Figure 5.7. Error images of all methods for 100th frame of 3rd video 65

Figure 5.8. 2AFC test and the decisions on Excel sheet 67

Figure 5.9. Structure of the preference vector 68

Figure 5.10. Histogram of Hamming distances for 28 preference vectors 69

Figure 5.11. Block diagram of outlier detection stage 70

Figure 5.12. Computation of preference scores for video and still frames 71

Figure 5.13. Computation of Spearman's rank correlation coefficients between subjective video and subjective frame stills 74

Figure 5.14. Computation of Spearman's rank correlation coefficients between subjective scores and objective measurements 76

Figure 5.15. Extraction of subvectors for each "method A vs. method B" comparison 77

LIST OF TABLES

Table 5.1. PSNR values of all methods for three different video sequences 57

Table 5.2. SSIM values of all methods for three different video sequences 57

Table 5.3. PSNR values of all methods for six different frame stills 62

Table 5.4. SSIM values of all methods for six different frame stills 62

Table 5.5. Rank and subjective score of each method for video sequences 72

Table 5.6. Rank and subjective score of each method for frame stills 72

Table 5.7. Spearman's rank correlation coefficients between subjective video scores and subjective frame scores for each comparison 74

Table 5.8. Spearman's rank correlation coefficients between subjective scores and PSNR measurements 78

Table 5.9. Spearman's rank correlation coefficients between subjective scores and SSIM measurements 78

LIST OF SYMBOLS / ABBREVIATIONS

f	Sampled version of degraded low resolution input image
f_c	Continuous version of degraded low resolution input image
g	Sampled version of original high resolution image
g_c	Continuous version of original high resolution image
$\{f\}$	Sampled version of degraded low resolution input image sequence
$\{f_c\}$	Continuous version of degraded low resolution input image sequence
$\{g\}$	Sampled version of reconstructed high resolution image sequence
$\{f_c\}$	Continuous version of reconstructed high resolution image sequence
(m,n)	Pixel coordinates on 2D discrete image plane
(x,y)	Pixel coordinates on 2D continuous image plane
(u,v)	Frequency coordinates on 2D continuous frequency plane
t	Iteration number
T	Sampling period
T_x	Sampling period on x direction
T_y	Sampling period on y direction
ω_x	Sampling frequency on x direction
ω_y	Sampling frequency on y direction
t_x	Shift on x direction
t_y	Shift on y direction
p_h	High resolution pixel
p_l	Low resolution pixel
G	DFT of g
G_c	CFT of g_c
F	DFT of f
F_c	CFT of f_c
η	Additive noise
ψ	2D interpolation kernel
ϕ	1D interpolation kernel
$\lfloor x \rfloor$	Greatest integer that is less than x

CFA	Color Filter Array
CFT	Continuous Fourier Transform
DFT	Discrete Fourier Transform
DTFT	Discrete Time Fourier Transform
GGD	Generalized Gaussian Distribution
GGMRF	Generalized Gauss Markov Random Field
GMRF	Gauss Markov Random Field
HMRF	Huber Markov Random Field
HR	High Resolution
IQR	Interquartile Range
LMS	Least Mean Squares
LR	Low Resolution
LSI	Linear Space Invariant
LSTV	Linear Space and Time Variant
LSV	Linear Space Varying
MAP	Maximum a Posteriori
ML	Maximum Likelihood
MRF	Markov Random Field
MSE	Mean Square Error
POCS	Projection onto Convex Sets
PSF	Point Spread Function
PSNR	Peak Signal to Noise Ratio
SR	Super Resolution
SSIM	Structural Similarity
2AFC	Two Alternative Forced Choice
2D	Two Dimensional
3D	Three Dimensional

1. INTRODUCTION

Digital imaging systems including all from primitive camcorders to scientific purpose cameras sample a real world scene by its sensor chip. If we model this system as a pinhole camera then the 3D scene is projected onto a 2D plane. The light coming from real world scene passes through camera optical system before reaching camera sensor.

During image acquisition process, the projected version of the 3D scene undergoes some degradation. What we see at the displays is the degraded version of the original real world scene. The limitations of the imaging system corrupts captured image. These limitations can be inspected under three category which are blur, aliasing and noise. Inefficiency of the optical system and Point Spread Function of sensor chip cause blur. Also temporal subsampling of the scene causes motion blur in video sequences. So capturing a single frame from a video sequence generally gives poor performance because of the blur around boundaries of moving objects.

The sensor chip is a 2D matrix composed of small photosensors called pixels. These sensors output data proportional to light amount (number of photons) falling on each of them. The image is formed fusing three monochromatic images red, green and blue. In ideal case, every pixel is composed of three subpixels for each monochromatic component. But in practice this is not the case. To reduce cost, every pixel is designed to measure only one color component. A Color Filter Array (CFA) is placed in front of the sensor array. So the light falling on each pixel is filtered according to the color pattern on CFA. A well known CFA pattern is Bayer pattern, taking its name from its inventor, Dr. Bryce E. Bayer of Eastman Kodak. This subsampling in the sensor array causes aliasing in the captured image. Finally the noise, which is formed on sensor array or during the transmission of the image data, corrupts the captured image. SR image reconstruction methods aim to create an HR image using LR image (or images).

1.1. Single Frame SR Image Reconstruction

The methods in this group use a database (or look-up-table) to reconstruct plausible high frequency components for the resulting HR image. HR image patches corresponding to different LR image patches are placed in this database. Database is created via a training phase using an independent set of training images. According to one exemplary training method [1], first, training images (original HR images) are smoothed and subsampled to model the degradation that LR images undergo. Then, degraded LR images are interpolated to the original HR image size using a standard analytical interpolation method (bilinear, bicubic or a similar method). At this stage, we have the original HR image and the corresponding interpolated LR image, namely, we have HR image patches corresponding to degraded LR image patches.

All of these matched patches are stored in our database and same procedure is repeated for all training images. This means a huge amount of data to store. Principle Component Analysis (PCA) can be used for dimensionality reduction. Also, it is known that high frequency components have the dominant role in predicting the details of an image. Another method to reduce database size is to apply highpass filtering to both interpolated LR image and original HR image and to store patches only for these high frequency components. Database can be implemented in a tree format to facilitate the search.

After creating database, we can construct an HR image from a single LR image by first interpolating LR image to the size of HR image and then predicting high frequency components using the information recorded in our database. Another issue related to Single Frame SR Image Reconstruction method is the spatial consistency of the high frequency patches found for neighboring pixels. The consistency of the neighboring patches should be taken into account at the database search process to get plausible high frequency components throughout the image.

This training based reconstruction method is modified for the case of multi frame (video) applications in [2].

1.2. Multi Frame SR Image Reconstruction

The methods in this group fuse data coming from different LR images of the same scene. These LR images are different looks of the same scene and they can be obtained from multiple cameras located at different locations or a single camera can be used to grab different looks of the scene. There should be relative subpixel shifts between LR images in order to use them in SR image reconstruction process. If there are integer pixel shifts, then these LR images contain the same information and can not be used for SR reconstruction.

After getting LR images, reconstruction process is established in three stages: registration (or motion estimation), interpolation onto an HR grid and image restoration [3]. In registration stage, subpixel shifts between LR images are estimated and LR images are matched spatially. So, same physical regions at two different LR images correspond to same spatial coordinates. These registered images are placed onto a global grid (HR grid). After placing onto HR grid, unavailable pixels are interpolated using the data coming from all LR images. Last stage is image restoration which is a well studied subfield of image processing. At this stage deblurring and denoising is applied to reconstructed HR image. Multi frame SR Reconstruction methods can be inspected in two broad classes as Frequency Domain Methods and Spatial Domain Methods.

Spatial Domain Methods:

- Interpolation of Non-Uniformly Spaced Samples
- Simulate and Correct Methods
- Probabilistic Methods
  - MAP Reconstruction Methods
  - ML Reconstruction Methods
- Set Theoretic Methods
  - Projection Onto Convex Sets
  - Bounding Ellipsoid Method
- Adaptive Filtering Methods
- Tikhonov-Arsenin Regularized Methods
- Robust Super Resolution

Frequency Domain Methods:

- Reconstruction via Alias Removal
- Recursive Least Squares Techniques

1.3. Outline of the Thesis

In Section 2, super resolution reconstruction methods are covered in detail. Section 3 gives information about common analytical interpolation methods. In Section 4, image and video quality metrics are given. In Section 5, results are given and evaluated. Finally, Section 6 is the conclusion part.

2. SUPER RESOLUTION RECONSTRUCTION METHODS

In this chapter, the theory of most commonly used SR reconstruction methods are given in detail.

2.1. Interpolation of Non-Uniformly Spaced Samples

This is the most intuitive method for SR image reconstruction. The aim is to reconstruct HR images from degraded LR image sequences. It is basically composed of three stages registration (motion estimation), nonuniform interpolation and restoration (deblurring and denoising) as shown in the following figure.

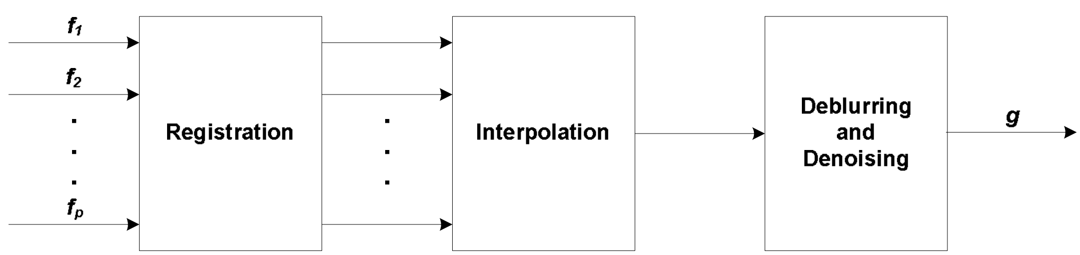


Figure 2.1. Super resolution stages

f_i is the degraded LR image and g is the reconstructed HR image. In registration stage, relative motion between LR images is estimated with subpixel accuracy. The accuracy of subpixel motion estimation is a very important factor affecting the performance of the SR reconstruction process. After finding relative shifts, LR images are placed on a uniform HR grid. Integer pixel shifts do not provide information. In Figure 2.2, red squares have integer pixel shifts. On the other hand, blue and yellow squares have subpixel shifts. At the end of registration stage, the result is nonuniformly spaced samples of registered LR images.

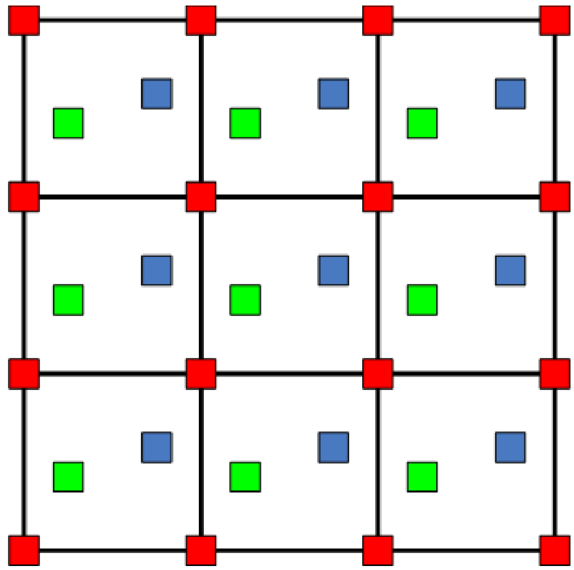


Figure 2.2. Subpixel shifts

In interpolation stage, the values of the uniformly spaced samples on the HR grid are interpolated using the values of nonuniformly spaced samples. Various methods exist in the literature for nonuniform interpolation.

Saito, Komatsu and Aizawa [4], [5], [6] proposed an interpolation method based on Landweber algorithm [7]. They used multiple LR cameras to reconstruct high resolution images. In registration stage, they used block matching technique to estimate relative shifts between LR images. Input images are interpolated *N* times, blocks of magnified images are compared and displacement of blocks are estimated with *1/N* subpixel accuracy. After registration, they used iterative Landweber algorithm in the interpolation stage to calculate uniformly spaced sample values from nonuniformly spaced samples. If the continuous image function g_c is approximated as a bandlimited signal and represented with its uniformly spaced samples as g, then its nonuniformly spaced samples f are related to its uniformly spaced samples g through the equation.

$$f = Ag \tag{2.1}$$

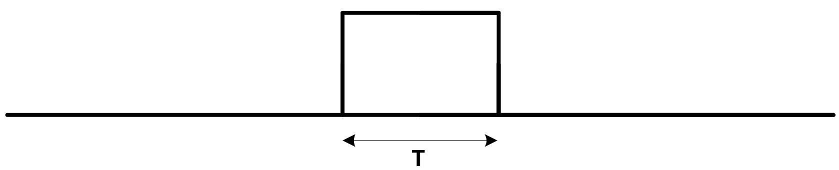
Here A is a linear matrix operator representing nonuniform sampling process and blur caused by image acquisition system. Next step is to estimate uniformly spaced samples g. Since the size of A matrix is proportional to the number of samples, calculating Moore-

Penrose pseudo inverse is a very computationally expensive operation. Instead, Landweber algorithm can be used. According to Landweber algorithm, g at iteration $(t+1)$ is related to g at iteration (t) through the relation

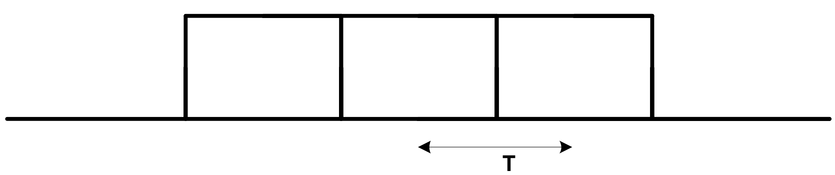
$$g^{(t+1)} = g^{(t)} + T(f - Ag^{(t)}) \tag{2.2}$$

where operator T is a mapping from the space F of nonuniformly spaced samples to space G of uniformly spaced samples. As the operator T, Saito, Komatsu and Aizawa used adjoint operator A^*. They also used a parameter α to control the process. Also $g^{(0)}$ can be chosen arbitrarily.

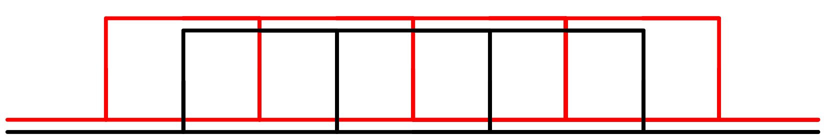
$$g^{(t+1)} = g^{(t)} + \alpha A^*(f - Ag^{(t)}) \tag{2.3}$$



(a)



(b)



(c)

Figure 2.3. Pixel apertures

The achievable passband of a CCD imager depends on the impulse response of the pixel aperture. The aperture effect limits the performance of the imaging system. Figure 2.3-a shows a single aperture (single pixel). Assuming a square pixel aperture, its spectrum is simply a sinc function. Figure 2.3-b shows a 1D array of pixels in the case of 100% aperture ratio. The sampling interval is T so our sampling frequency is $1/T$. Sinc functions overlap on frequency domain so the bandwidth of the input signal should be limited under $1/2T$ (half of the sampling frequency). This result is consistent with well known Nyquist theorem.

If we use two CCD imagers to capture the same scene, we can increase the performance of our system. Two CCD imagers are placed in a special geometric arrangement so that their sampling arrays overlap as shown in Figure 2.3-c. One imager samples the scene as shown by black lines and the other samples the scene by red lines. So we increase our sampling frequency to $2/T$ and no degradation occurs for frequencies under $1/T$. This means we can increase the performance of the system to the limitation set by single pixel aperture.

In [4], [5] Saito, Komatsu and Aizawa used multiple cameras with the same pixel apertures to construct HR images. When using cameras with the same pixel aperture, the resolution improvement depends on the geometric configuration of the scene and the cameras. The cameras should be coplanar and 2D image plane should be perpendicular to their optical axes. In [8] Saito, Komatsu and Aizawa proposed using multiple cameras with different pixel apertures in image acquisition process. In the case of different apertures, pixel of one imager does not fully overlap with the pixel of another imager (refer to Figure 2.4). This way, pixel of each imager samples extra information without a special geometric configuration of the scene. The problem of this method is different aliasing artifacts of each imager. Since, aliasing depends on pixel aperture of each camera, different pixel apertures mean different aliasing effects for each camera. This problem reduces the system performance on registration stage.

To remedy this problem an alternately iterative method is proposed. In this method, registration and reconstruction stages are handled together. First, an improved resolution image is created using LR images. Then, registration procedure is repeated taking improved resolution image as reference. This time, relative shifts between LR images and improved resolution image are estimated. This way, the robustness of the registration stage is increased because reference improved resolution image contains most of the information coming from all LR images.

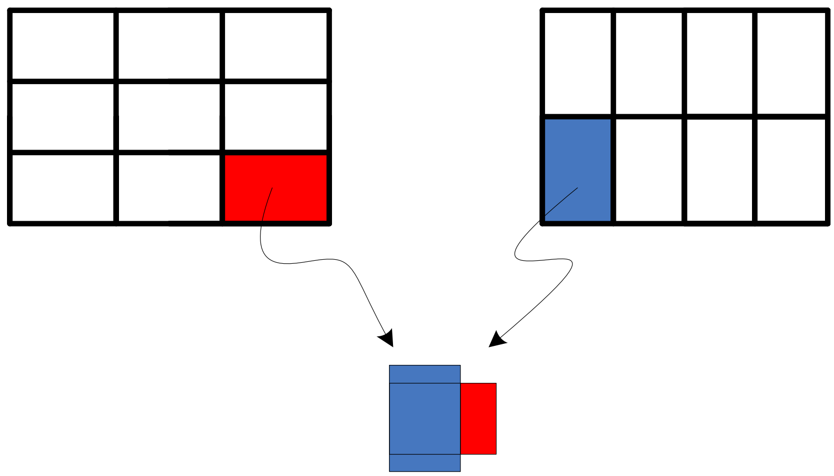


Figure 2.4. Different Pixel Apertures

In [9] and [10], Nakazawa, T. Saito, T. Sekimori, and K. Aizawa used a temporal integration method to increase spatial resolution using a sequence of images. It is an application oriented approach. The images are captured using a low resolution camera moving along a track. A surveillant determines a ROI on observed low quality image. Then temporal integration algorithm combines the information coming from low quality images to improve the resolution inside the ROI. The method mainly consists of two stages: motion estimation and nonuniform interpolation.

For motion estimation they proposed a special algorithm named as quadrilateral motion estimation. In quadrilateral motion estimation, first the ROI is divided into quadrilateral patches as shown in Figure 2.5. The method describes image motion as the deformation of these patches from one frame to another (refer to Figure 2.6). The deformation can be modeled using perspective projection. Note that affine transformation has six degrees of freedom and is not enough in this case. We can model only triangle to triangle deformations using affine transformation. To model quadrilateral to quadrilateral deformation, we need eight degrees of freedom which means perspective projection. After finding the parameters of perspective projection, we can estimate the motion of each pixel inside quadrilateral patches.

In order to estimate projection parameters, the grid points on HR grid are moved in small amounts on horizontal and vertical directions. MSE is estimated between reference ROI and the ROI under consideration after each movement. The movement minimizing MSE determines the correct position of grid point in successive frames (refer to Figure 2.7).

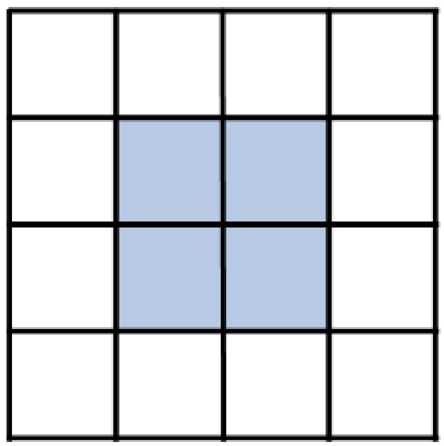


Figure 2.5. ROI covered with quadrilaterals

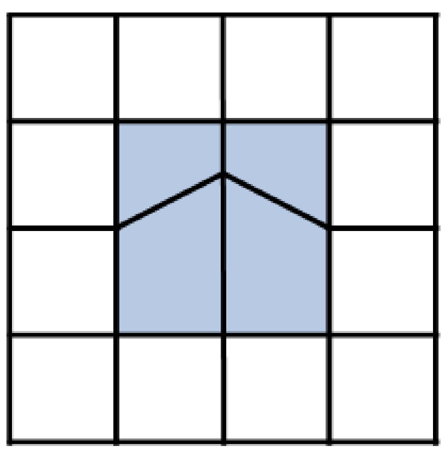


Figure 2.6. Deformation of quadrilaterals

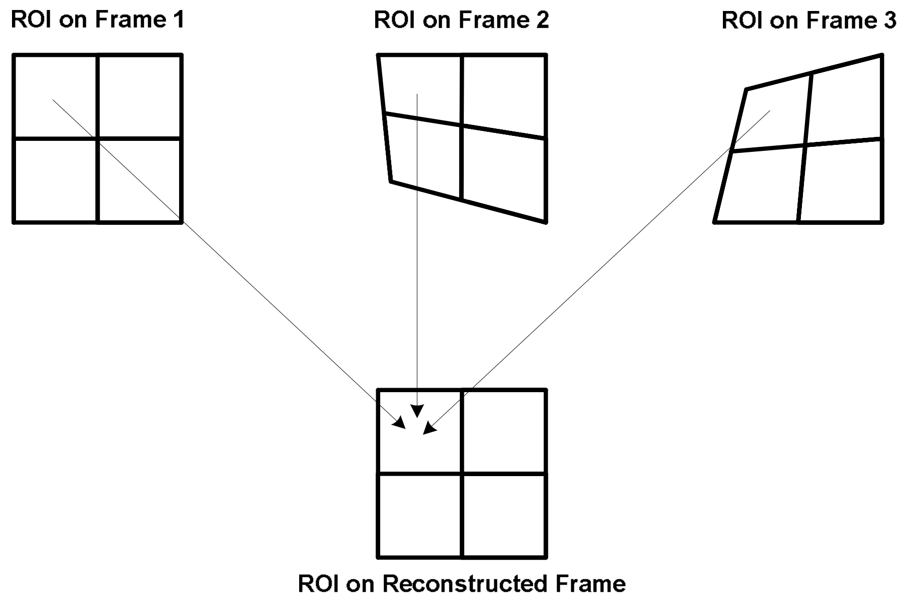


Figure 2.7. Quadrilateral patches and deformation

After registration, next step is to make interpolation using nonuniformly spaced samples. An HR grid is determined inside the undeformed ROI. Each luminance value (p3) on this HR grid is estimated as weighted average of nonuniformly spaced pixel values (p1 and p2) around HR pixel. Figure 2.8 illustrates this operation. Nonuniformly spaced pixels used in weighted averaging are within a chess-board distance of one HR pixel.

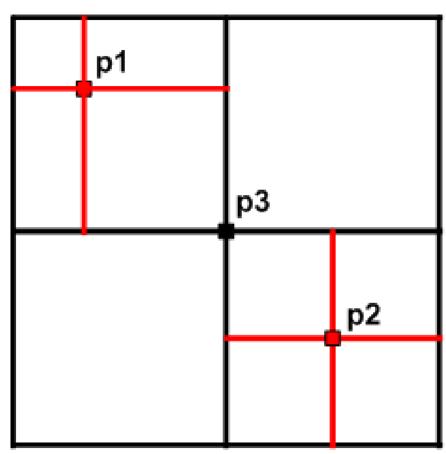


Figure 2.8. Nonuniform interpolation

After this operation, some pixel values remain undecided. An iterative approach is used to find undecided pixel values. The equation of the method is given below

$$g_{m,n}^{(t+1)} = g_{m,n}^{(t)} + \omega\left(g_{m+1,n}^{(t)} + g_{m-1,n}^{(t)} + g_{m,n+1}^{(t)} + g_{m,n-1}^{(t)} - 4g_{m,n}^{(t)}\right) \tag{2.4}$$

$g_{m,n}$ is the intensity value of a HR pixel at *(m,n)* position and t is iteration number. This formula is used iteratively until convergence. ω is used to control the rate of convergence.

Keren, Peleg and Brada [11] used a temporal averaging method to obtain uniformly spaced HR samples. As the first step, LR images are registered and placed onto an HR grid. Each sample in these LR images (size of *NxN*) has an area of *dxd*. On HR image (size of *MNxMN*), determine an area of *d/Mxd/M* around each unknown sample and call this area as q. All LR images are on top of each other on HR grid. Assume that you push a needle through the center of q. This needle will pierce a sample area in each LR image. Call these samples corresponding to pierced areas as p^k. Averaging all p^k values we get the value of HR sample at the center of q. As a result of averaging process, the result is a blurred version of the original image. Deblurring or high-pass filtering can be applied to enhance high frequency details.

2.2. Simulate and Correct Methods

In simulate and correct methods, first an initial estimate of HR image is produced. Then a transform modeling the imaging process is applied to this estimate and LR images are obtained. These LR images are compared with original LR images. The error is then used to modify initial HR estimate and this operation continuous iteratively.

In [12] and [13], Shah and Zakhor proposed a multiframe resolution enhancement technique. The method consists of three stages. First, registration is performed and a motion vector is found for each pixel in LR images relative to the reference LR image. They proposed to use 5 LR images {k-2, k-1, k, k+1, k+2} and used kth image as the reference. In registration stage they used Modified Block Matching Algorithm (MBMA). In MBMA, a candidate set of motion vectors is found for each pixel using MSE as the comparison criterion. For each pixel, the motion vectors close to the best motion vector (lowest MSE) less than a predetermined threshold are accepted as candidate vectors. The reason to use a set of candidate motion vectors instead of one is to increase the motion estimation accuracy. According to empirical results, due to aliasing and blurring, true motion vector can sometimes have higher error than best motion vector in terms of MSE.

After registration, the resulting motion vectors are used to combine LR images into an initial HR estimate. Since we have more than one motion vector for each pixel, several situations can arise. If a single LR pixel is mapped to an HR pixel, then the value of this HR pixel becomes the value of the LR pixel. If multiple LR pixels are mapped to an HR pixel, then the pixel with motion vector of lowest MSE is selected. This way, most of the pixels on HR grid are found. The values of the remaining holes are found using iterative Landweber algorithm. First, the imaging process is simulated on the initial estimate of HR image using motion information for different LR images. The resulting LR images are then compared with original LR images. Then, the error is used to modify initial estimate of HR image. This process continues iteratively until convergence.

$$g^{(t+1)} = g^{(t)} + \alpha A^{*}(f - Ag^{(t)}) \tag{2.5}$$

Above equation describes the process for each LR image. In this case, f is the original LR image. $g^{(t)}$ is the estimate of HR image at iteration t. Imaging is simulated on $g^{(t)}$. The error between the original LR image and simulated LR image is projected to the space of g using A^{*}. This error is added to $g^{(t)}$. At each iteration, $Ag^{(t)}$ gets closer to y.

In [14], Irani and Peleg assumed only global translational and global rotational motion to describe motion from frame to frame. The imaging process is modeled by

$$f_k(m,n) = Q_k(h(g_c(x,y)) + \eta_k(x,y)) \tag{2.6}$$

where f_k is the kth observed digitized LR image, g_c is the original continuous HR image, h is the blurring operator and η_k is additive noise term for kth observed image. Q_k samples and quantizes the blurred and noisy version of HR input image. It also includes the effect of image warping due to motion. (x,y) is the center of the receptive field of the detector whose output is $f_k(m,n)$. (x,y) and (m,n) are related to each other as shown below:

$$x = t_x + T_x m\cos\theta_k - T_y n\sin\theta_k \tag{2.7}$$

$$y = t_y + T_x m \sin\theta_k + T_y n \cos\theta_k \tag{2.8}$$

where T_x and T_y are the sampling periods, t_x and t_y are translation parameters and θ_k is the rotation angle of the *kth* observed image. If we use above motion parameters, we can write a relationship between two consecutive frames f_1 and f_2 as:

$$f_2(x, y) = f_1(x\cos\theta - y\sin\theta + t_x, y\cos\theta + x\sin\theta + t_y) \tag{2.9}$$

Irani and Peleg found motion parameters by solving this equation. Detailed derivation can be found in [14]. The method to find motion parameters is used iteratively to increase the accuracy of motion parameters. First, initial estimates of motion parameters are found. Then image f_2 is warped thorough image f_1. Motion parameters are calculated again and added to initial estimates. Accuracy of the motion parameters increases at every iteration.

Knowing motion parameters, the imaging process can be simulated on HR estimate and the resulting LR images can be compared to observed LR images. Denote original discrete HR image as g, HR pixel as p_h and LR pixel as p_l. First, an initial estimate $g^{(0)}$ of HR image is computed. After applying imaging process to $g^{(0)}$ we get simulated LR images $f_k^{(0)}$ as shown below:

$$f^{(t)}(p_l) = \sum_{p_h} g^{(t)}(p_h) h_1(p_h - z_{pl}) \tag{2.10}$$

where z_{pl} represents the center of receptive field of LR pixel p_l. The value of the LR pixel p_l is calculated by weighted averaging of the HR pixels in its receptive field. The weight of HR pixels increases as the HR pixel location approaches to z_{pl}. Here h_1 is a smoothing kernel. Convolving HR image estimate $g^{(t)}$ with h_1 we get the simulated LR image $f^{(t)}$. After finding simulated LR images, $g^{(t)}$ is updated using the error between simulated LR images and observed LR images.

$$g^{(t+1)}(p_h) = g^{(t)}(p_h) + \sum_{p_l \in L_k} (f_k(p_l) - f_k^{(t)}(p_l)) h_2(p_h - p_l) C \tag{2.11}$$

where L_k is the set of LR pixels that are influenced by p_h (p_h is in the receptive fields of all LR pixels in L_k). The effect of h_2 is to weight the contribution of the error terms. In this way, the weight of the error coming from a LR pixel is high if the center of receptive field of that LR pixel gets closer to p_h. h_2 can be set equal to h_1 or another filter can be used. C is normalization constant. Weighted averaging the error terms coming from different neighbor pixels decreases additive noise strength. In order to handle multiplicative noise, median filtering can be applied to difference image $(f_k - f_k^{(t)})$ before averaging the errors.

2.3. MAP and ML Reconstruction Methods

The SR image reconstruction problem is treated in a statistical framework. In ML estimation, likelihood of HR estimate with respect to observed LR image (or images) is maximized. The block diagram of the process is illustrated in Figure 2.9. A priori information about HR estimate is incorporated in MAP estimation. We can say that ML estimate is also a MAP estimate with a flat prior for the HR image.

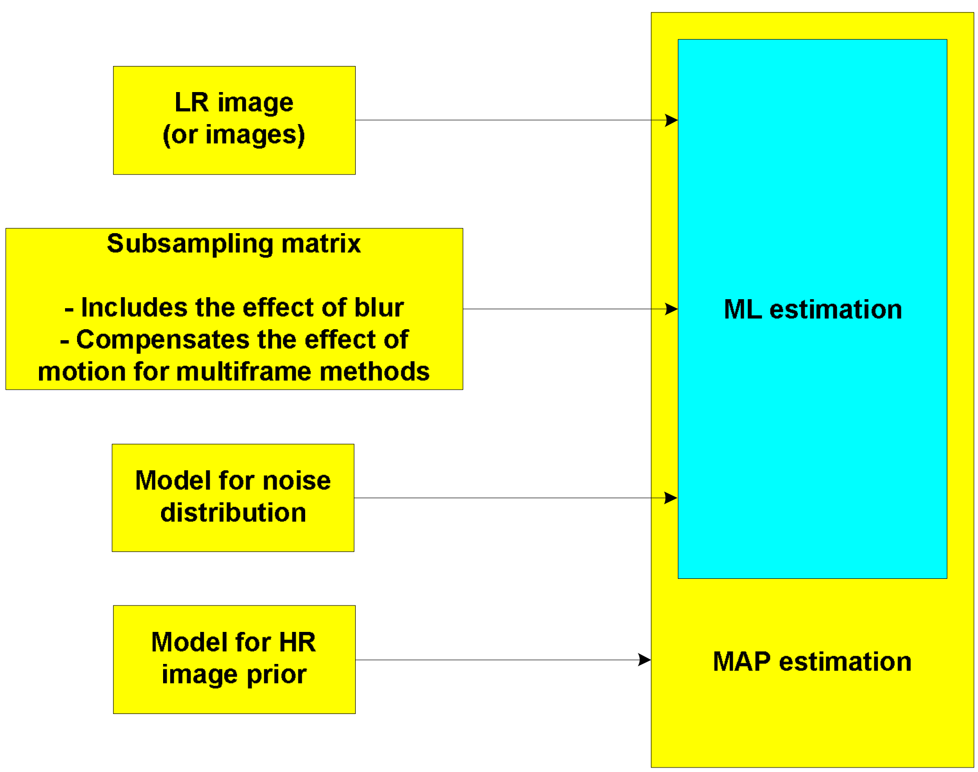


Figure 2.9. Block diagram for MAP and ML estimation of HR image

Schultz and Stevenson [15], [16] proposed a Bayesian approach for noiseless and noisy images. First the image acquisition process is modeled. Let $g_c(x,y,t)$ be the continuous HR image. Then f is the digitized version of g_c.

$$f\left(m,n\right)=\int_0^1\int_{wm}^{w(m+1)}\int_{wn}^{w(n+1)} g_c(x,y,t)dxdydt \tag{2.12}$$

Here $f(m,n)$ is the pixel value at the (m,n) position where $m=0,\ldots.,N_1-1$ and $n=0,\ldots.N_2-1$. w is the side length of the square pixel aperture. Simply a local neighborhood is integrated to determine the value of the pixel at the corresponding position of the grid. If we want a higher resolution image then we should use a finer grid during image acquisition.

$$g(k,l)=\int_0^{q^2}\int_{\frac{wk}{q}}^{\frac{w(k+1)}{q}}\int_{\frac{wl}{q}}^{\frac{w(l+1)}{q}} g_c(x,y,t)dxdydt \tag{2.13}$$

Here, $k=0,\ldots.,qN_1-1$ and $l=0,\ldots,qN_2-1$. In this case, side length of pixel aperture decreased to w/q and g has dimension qN_1xqN_2. Then the relation between low resolution image f and high resolution image g is

$$f(m,n)=\frac{1}{q^2}\left(\sum_{k=qm}^{q(m+1)-1}\sum_{l=qn}^{q(n+1)-1} g(k,l)\right) \tag{2.14}$$

f is the blurred version of g according to this model. If f and g are written in lexicographical notation, the relationship can be interpreted in another way.

$$f=Dg \tag{2.15}$$

Here f is N_1N_2x1, g is $q^2N_1N_2x1$ and D is $N_1N_2x\ q^2N_1N_2$. If noise is included in the model, the equation becomes

$$f=Dg+\eta \tag{2.16}$$

A maximum a posteriori (MAP) technique is proposed to obtain g from f.

$$\hat{g} = \arg\max_{g} L(g \mid f) \tag{2.17}$$

L(g|f) is the log-likelihood function (since log function is monotonically increasing, the parameters maximizing log-likelihood function also maximizes likelihood function). Using Bayes' formula

$$\begin{aligned} L(g \mid f) &= \log P(g \mid f) \\ &= \log P(f \mid g) + \log P(g) - \log P(f) \end{aligned} \tag{2.18}$$

Third term on the right can be ignored in the optimization because it is not a function of g. Then MAP estimate becomes

$$\begin{aligned} \hat{g} &= \arg\max_{g} \{\log P(f \mid g) + \log P(g)\} \\ &= \arg\min_{g} \{-\log P(f \mid g) - \log P(g)\} \end{aligned} \tag{2.19}$$

Here the densities *P(f|g)* and *P(g)* should be identified to find the estimate. For noiseless images

$$P(f \mid g) = \begin{cases} 0, & f \neq Dg \\ 1, & f = Dg \end{cases} \tag{2.20}$$

And for noisy images,

$$P(f \mid g) = P(\eta)\,|_{\eta = f - Dg} \tag{2.21}$$

If we model the noise as additive white Gaussian noise then,

$$P(f \mid g) = \frac{1}{(2\pi\sigma^2)^{\frac{N_1 N_2}{2}}} \exp\left(\frac{-\|f - Dg\|^2}{2\sigma^2}\right) \tag{2.22}$$

where σ^2 is noise variance. Finally image is modeled as a Markov Random Field with Gibbs density function.

$$P(g) = \frac{1}{Z}\exp\left(-\frac{1}{\lambda}\sum_{c\in C} V_c(g)\right) \tag{2.23}$$

Using these densities our estimate for noiseless case becomes

$$\hat{g} = \arg\min_{g\in Z}\left\{\frac{1}{\lambda}\sum_{c\in C} V_c(g)\right\} \tag{2.24}$$

and for noisy case

$$\begin{aligned}\hat{g} &= \arg\max_{g}\left\{\log\left(\frac{1}{(2\pi\sigma^2)^{\frac{N_1N_2}{2}}}\right) - \frac{\|f-Dg\|^2}{2\sigma^2} + \log\left(\frac{1}{Z}\right) - \frac{1}{\lambda}\sum_{c\in C} V_c(g)\right\} \\ &= \arg\min_{g}\left\{\frac{\|f-Dg\|^2}{2\sigma^2} + \frac{1}{\lambda}\sum_{c\in C} V_c(g)\right\}\end{aligned} \tag{2.25}$$

In order to obtain a global minimum, Huber function (a convex function) is used in the definition of V_c. Huber function ρ, as illustrated in Figure 2.10, is quadratic below a threshold, and linear above the threshold.

$$\rho_T(x) = \begin{cases} x^2, & |x| \le T \\ T^2 + 2T(|x| - T), & |x| > T \end{cases} \tag{2.26}$$

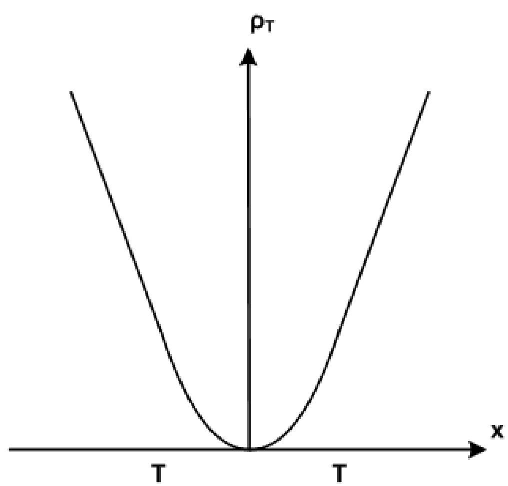


Figure 2.10. Huber function with threshold T

Using Huber function V_c becomes

$$\sum_{c\in C} V_c(g) = \sum_{k=0}^{qN_1-1} \sum_{l=0}^{qN_2-1} \sum_{m=0}^{3} \rho_T(d_{k,l,m}^t g) \tag{2.27}$$

Using derivative approximations for the discrete case, we can define $d_{k,l}$ as

$$\begin{aligned}
d_{k,l,0}^t g &= g_{k,l+1} - 2g_{k,l} + g_{k,l-1} \\
d_{k,l,1}^t g &= \frac{1}{2}(g_{k-1,l+1} - 2g_{k,l} + g_{k+1,l-1}) \\
d_{k,l,2}^t g &= g_{k-1,l} - 2g_{k,l} + g_{k+1,l} \\
d_{k,l,3}^t g &= \frac{1}{2}(g_{k-1,l-1} - 2g_{k,l} + g_{k+1,l+1})
\end{aligned} \tag{2.28}$$

This set of $d_{k,l}$ and Huber function protects strong discontinuities while blurring weak details.

The g image minimizing our estimate is found using gradient projection algorithm (constrained optimization) for the noiseless case and gradient descent algorithm (unconstrained optimization) for the noisy case [15].

In [17], [18] and [19], Schultz and Stevenson modified the single frame method described in [15] and [16] to include temporal information coming from neighboring images. In multiframe case, the only difference is that the decimation matrix also compensates for the motion between LR frames.

In [20], Bouman and Sauer used generalized Gaussian Markov random field (GGMRF) to model the image prior and noise. Gaussian prior generally gives a smooth image (blurry edges), because the edges are penalized with the square of the edge magnitude. A better model should permit discontinuities. So the tails of the distribution should be longer than Gaussian distribution. The generalized Gaussian distribution (GGD) is a good choice, because using a shape parameter, tail length of the distribution can be controlled. The imaging process is modeled as

$$f = Ag + \eta \tag{2.29}$$

where η is the noise vector. Elements of η are i.i.d.. The distribution of η can be modeled using generalized Gaussian distribution

$$p(\eta) = \frac{q}{2\Gamma\left(\frac{1}{q}\right)} \exp\left(-|\eta|^{q}\right) \tag{2.30}$$

where Γ is gamma function and q is the shape parameter. If $q=2$, then distribution is Gaussian. If it is 1, then distribution is Laplacian. As q decreases from 2, the distribution becomes more heavy-tailed.

In [21], Hardie, Barnard and Armstrong estimated an HR image from a sequence of LR images using MAP estimation. An imager is assumed to be mounted on a vehicle or an aircraft. The movement of the vehicle provides a jitter and subpixel motion occurs between successive LR images. This subpixel motion is exploited during SR image reconstruction. In [21], only global translational motion is taken into account but it is said that the algorithm is applicable for a more general scenario. The HR image prior and noise are modeled as Gauss-Markov random fields.

In [22] and [23], Elad and Feuer incorporated the methods used for single image restoration to the more complex problem of super-resolution image reconstruction. ML method to restore one image from a single degraded image is generalized to the problem of single image restoration from several measured images.

The first step is the formulation of the problem. There are N measured low resolution, degraded images f_k at hand ($k=1,\ldots,N$). Each image has different size $M_k x M_k$. Our aim is to create a super-resolution image g with dimension LxL, fusing the information of measured images. Each measured image is produced applying warping, blurring and decimation to the high resolution image g. Also, the measured images are degraded with additive noise. Noise distribution will be modeled as Gaussian. For the sake of generality, it is assumed that different warping, blur, noise and decimation are applied to each measured image. This model can be expressed analytically as shown below:

$$f_k = D_k C_k F_k g + \eta_k \text{ for } 1 \le k \le N \tag{2.31}$$

In this configuration, high resolution image g and low resolution measured images f_k are shown in lexicographical notation. F_k represents geometric warping and has a size of $L^2 x\, L^2$. C_k is the linear space-variant blur matrix of size $L^2 x\, L^2$. D_k is the decimation matrix of size $M_k^2 x L^2$. Finally η_k represents additive zero mean Gaussian noise. All these matrices are assumed to be known. Writing the above equation for all measured images and expressing the result using matrix notation we get:

$$\begin{bmatrix} f_1 \\ \vdots \\ f_N \end{bmatrix} = \begin{bmatrix} D_1 C_1 F_1 \\ \vdots \\ D_N C_N F_N \end{bmatrix} g + \begin{bmatrix} \eta_1 \\ \vdots \\ \eta_N \end{bmatrix} = \begin{bmatrix} H_1 \\ \vdots \\ H_N \end{bmatrix} g + N \tag{2.32}$$

$$\Phi = Hg + N \tag{2.33}$$

where $H_k = D_k C_k F_k$ and the autocorrelation matrix of Gaussian random vector N is W^{-1}. Block diagram of the model is shown in Figure 2.11.

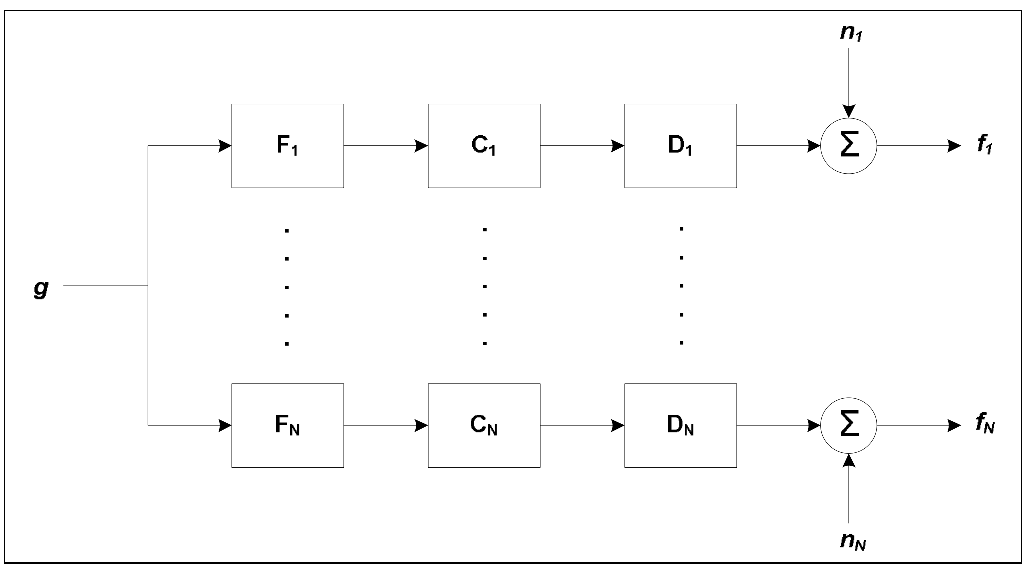


Figure 2.11. Block diagram of the system

According to ML estimation, high resolution image can be found by maximizing the conditional probability density function of the measurements given the ideal image, $P(\Phi \mid g)$.

$$\begin{aligned}\hat{g}_{ML} &= \arg\max_{g} P(\Phi \mid g) \\ &= \arg\max_{g}\left([\Phi - Hg]^T W[\Phi - Hg]\right)\end{aligned} \tag{2.34}$$

To minimize the expression on the right side, we can take the derivative with respect to g and equate to zero. The result is

$$H^T W H \hat{g}_{ML} = H^T W \Phi \tag{2.35}$$

Locally adaptive regularization can also be included in the equation. Using a Laplacian operator S and a weighting matrix V, we get

$$\hat{g}_{ML} = \arg\max_{g}\left([\Phi - Hg]^T W[\Phi - Hg] + \beta[Sg]^T V[Sg]\right) \tag{2.36}$$

2.4. Projection onto Convex Sets

Brief review of POCS method is given in [24]. Mathematically, POCS is the operation of finding a point which lies in a convex set and is nearest to another point generally not lying in the set. If we denote the image to be reconstructed as g, then every known property of g constrains g to lie in some convex set. For m properties of g, there are m convex sets. And the reconstruction problem is to find a point at the intersection of these convex sets.

In [25], Tekalp, Ozkan and Sezan proposed a POCS method incorporating the observation noise. Let $f_i(m,n)$ denote the sample of the *ith* low resolution image at *(m,n)* coordinates.

$$f_i(m,n) = \sum_{m=0}^{M-1}\sum_{n=0}^{M-1}\left[g_c(x-t_x, y-t_y) ** h(x,y)\right]\delta(x-mT, y-nT) + \eta_i(mT, nT) \tag{2.37}$$

Size of f_i is MxM . $g_c(x,y)$ denotes original continuous scene. t_x and t_y are the displacements of f_i with respect to reference frame in x and y directions, respectively. T is sampling period. $h(x,y)$ is the PSF of the sensor, $\eta_i(x,y)$ is additive noise component for ith frame and $**$ denotes 2D convolution. If continuous scene is sampled at a denser grid (with a sampling period smaller than T), then resulting high resolution image g and low resolution images f_i have a relationship as expressed below

$$f_i(m,n)=\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} g(k,l)h_i(m,n;k,l)+\eta_i(m,n) \tag{2.38}$$

High resolution image $g(k,l)$ has a size of NxN and h_i is the PSF for the ith frame. Following convex constraint is defined for each pixel of low resolution images

$$C_{m,n,i}=\left\{g(k,l):\left|r_i^{(g)}(m,n)\right|\leq\delta_0\right\}, \tag{2.39}$$

where $0\leq m,n\leq M-1,\ i=1,\ldots,L^2$ and

$$r_i^{(g)}(m,n)=f_i(m,n)-\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} g(k,l)h_i(m,n;k,l) \tag{2.40}$$

In this degradation model, observation noise is included, but aperture time is assumed to be zero. So, the effect of motion blur is neglected. The projection operator is given as.

$$P_{m,n,i}\left[g(k,l)\right]=\begin{cases} g(k,l)+\dfrac{r_i(m,n)-\delta_0}{\sum_o\sum_p h_i^2(m,n;o,p)}h_i(m,n;k,l) & if \quad r_i(m,n)>\delta_0 \\ g(k,l) & if \quad -\delta_0<r_i(m,n)<\delta_0 \\ g(k,l)+\dfrac{r_i(m,n)+\delta_0}{\sum_o\sum_p h_i^2(m,n;o,p)}h_i(m,n;k,l) & if \quad r_i(m,n)<-\delta_0 \end{cases} \tag{2.41}$$

In [26], Patti, Sezan and Tekalp proposed a POCS method to account for aliasing, sensor blur, motion blur and observation noise. In this new degradation model, motion blur due to sensor aperture time is modeled at the beginning. Same convex constraints are used in the high resolution image reconstruction process.

In [27], the work in [25] and [26] is extended to include an arbitrary spatio-temporal sampling lattice. Same observation model is used except that sampling stage is modified to handle arbitrary sampling lattices. POCS constraints given in [25] and [26] are used in reconstruction stage. Detailed version of the work in [25], [26] and [27] are given in [28] by Patti, Sezan and Tekalp.

2.5. Bounding Ellipsoid Method

In [29], Tom and Katsaggelos used a POCS method to reconstruct the HR estimate of the image using some neighbor images from the video sequence. They modeled the imaging process as described below.

Let g_k denote the kth lexicographically ordered high resolution frame of the sequence and f_k denote the degraded version of g_k. The low resolution frame can be obtained from its high resolution counterpart using a transformation matrix

$$f_k = A^{(k,k)} g_k \tag{2.42}$$

$A^{(k,k)}$ simulates the smoothing and decimation operations. A low resolution frame is of size NxN and a high resolution frame is of size $PNxPN$, where P is an integer. Then, $A^{(k,k)}$ is $N^2x(PN)^2$. For the neighbor images, the effect of motion should be taken into account.

$$f_{k-i} = A^{(k-i,k-i)} g_{k-i} = A^{(k,k-i)} g_k \tag{2.43}$$

The operator $A^{(k,k-i)}$ compensates motion from g_k to g_{k-i} and makes averaging and decimation. The effect of motion can be expressed explicitly as

$$A^{(k,k-i)} = A^{(k-i,k-i)} C^{(k,k-i)} \tag{2.44}$$

Here $A^{(k-i,k-i)}$ is responsible for smoothing and decimation. $C^{(k,k-i)}$ is the motion compensation operator from frame k to frame $k-i$. C is defined as

$$g_{k-i}(\mathbf{r}) = C^{(k,k-i)} g_k(\mathbf{r}) = g_k\left(\mathbf{r} + \mathbf{d}^{(k,k-i)}\right) \tag{2.45}$$

where $\mathbf{d}^{(k,k-i)}$ is the displacement vector field (DVF) between frames *k-i* and *k*.

Using these relations, a high resolution image g_k can be reconstructed from (M_1+M_2+1) low resolution images using an iterative algorithm.

$$g_k^{t+1} = g_k^t + \sum_{i=-M_1}^{M_2} \left(C^{(k-i,k)}\right)^T \left(A^{(k-i,k-i)}\right)^T r_{i,k} - \lambda Q^T Q g_k^t \tag{2.46}$$

where the residual term $r_{i,k}$ is

$$r_{i,k} = f_{k-i} - \beta_i A^{(k-i,k-i)} C^{(k,k-i)} g_k^t \tag{2.47}$$

λ is the regularization parameter, t denotes iteration number, Q is a high pass filter. The contribution of each low resolution frame is adjusted using β_i parameter. β_i is inversely proportional to amount of error ε in motion estimation which is given by

$$\varepsilon = \left| g_k(\mathbf{r}) - g_{k-i}\left(\mathbf{r} + \mathbf{d}^{(k,k-i)}\right) \right| \tag{2.48}$$

2.6. Tikhonov Arsenin Regularization

In [30], Hong, Kang and Katsaggelos proposed an iterative regularized algorithm to increase the resolution of a video sequence. g_i and f_i denote the HR image and corresponding LR image respectively. g_i and f_i have the same size.

$$f_i = D g_i + \eta_i \tag{2.49}$$

D is a spatially invariant PSF and models the smoothing effect on image acquisition system. η_i is the additive noise component. The problem is to estimate sequence of g_i given a sequence of f_i. Since the frames of a video sequence are temporally correlated, motion information will be used in the reconstruction process. Using motion compensation operator $U_{i,j}$, two frames of the sequence are related as shown below

$$g_i(m,n) = U_{i,j} g_j(m,n) \tag{2.50}$$

Considering one observation for m frames of the original HR video sequence

$$g = \left[g_k^T, g_{k+1}^T, \ldots, g_{k+m-1}^T \right]^T \tag{2.51}$$

$$f_i = \left[DU_{i,k}, DU_{i,k+1}, \ldots, DU_{i,k+m-1} \right] g + \eta_i \tag{2.52}$$

$$f_i = \left[H_{i,k}, H_{i,k+1}, \ldots, H_{i,k+m-1} \right] g + \eta_i = H_i g + \eta_i \tag{2.53}$$

Then, considering m observation, the HR estimate can be found by minimizing following regularized objective function

$$M(g) = \sum_{i=1}^{m} M_i(g) = \sum_{i=1}^{m} \left[\left\| f_i - H_i g \right\|_{A_i}^2 + \alpha_i \left\| C_i g \right\|_{B_i}^2 \right] \tag{2.54}$$

where

$$C_i = \left[C_{i,1}, C_{i,2}, \ldots, C_{i,m} \right] \tag{2.55}$$

$C_{i,j}$ is a matrix representing a high-pass filter and α_i is the regularization parameter for the ith frame. A_i and B_i are weighting matrices. Gradient of $M(g)$ is computed and equated to zero to find g that is minimizing $M(g)$.

$$\frac{\partial M(g)}{\partial g} = \sum_{i=1}^{m} \left[\left(H_i^T W_{A_i} H_i + \alpha_i C_i^T W_{B_i} C_i \right) g - H_i^T W_{A_i} f_i \right] \tag{2.56}$$

where

$$W_{A_i} = A_i^T A_i \quad \text{and} \quad W_{B_i} = B_i^T B_i \tag{2.57}$$

Then estimate of g can be found iteratively by using the expression below

$$g^{t+1} = g^t + \beta \sum_{i=1}^{m} \left[H_i^T W_{A_i} f_i - \left(H_i^T W_{A_i} H_i + \alpha_i C_i^T W_{B_i} C_i \right) g^t \right] \tag{2.58}$$

where g^t is the estimate at the iteration t.

2.7. Robust Super Resolution Reconstruction

In [31], Patanavijit and Jitapunkul proposed a robust SR algorithm. Their error function included two terms, one is for fidelity and the other is for regularization. Using single frame observation

$$Error = \rho\left(DHg - f\right) + \lambda\psi\left(\Gamma g\right) \tag{2.59}$$

where D and H are decimation and blur matrices, respectively. Lorentzian norms are defined as

$$\rho\left(x\right) = \log\left[1 + \frac{1}{2}\left(\frac{x}{T}\right)^2\right] \qquad \psi\left(x\right) = \log\left[1 + \frac{1}{2}\left(\frac{x}{T_g}\right)^2\right] \tag{2.60}$$

The g minimizing the error term is the estimate of the original HR image.

$$g = \arg\min_g \left\{\rho\left(DHg - f\right) + \lambda\psi\left(\Gamma g\right)\right\} \tag{2.61}$$

The solution can be found using steepest descent method

$$\hat{g}_{t+1} = \hat{g}_t + \beta\left\{H^T D^T \rho'\left(f - DH\hat{g}_t\right) - \left(\lambda\Gamma^T \psi'\left(\Gamma\hat{g}_t\right)\right)\right\} \tag{2.62}$$

where

$$\rho'\left(x\right) = \frac{2x}{2T^2 + x^2} \qquad \psi'\left(x\right) = \frac{2x}{2T_g^2 + x^2} \tag{2.63}$$

To find the fidelity term, the imaging process is simulated on SR estimate. Then, the difference between simulation result and the observation is found. This difference is modulated using Lorentzian norm ρ. Patanavijit and Jitapunkul used Lorentzian Laplacian regularization to increase the robustness against noise. Assuming a smooth prior, a highpass filter (Laplacian in this case) is applied to SR estimate and the result is modulated using Lorentzian norm ψ.

2.8. Adaptive Filtering (LMS) Method

In [33], [34], [35], [36] and [37] LMS filtering method is applied for SR reconstruction. In [33], Elad and Feuer proposed a super-resolution reconstruction algorithm for continuous image sequences. Let original image sequence be related to degraded image sequence via the relationship

$$f_c(t-k) = DH(t-k)F(t,k)g_c(t) + \eta(t,k) \tag{2.64}$$

where $\{f_c(t)\}_{t\geq 0}$ is degraded continuous image sequence of size MxM and $\{g_c(t)\}_{t\geq 0}$ is the ideal continuous image sequence of size LxL where $L \geq M$ and $0 \leq k \leq \infty$. D, $H(t\text{-}k)$ and $F(t,k)$ denotes decimation, LSTV blur and backward geometric warp operator respectively. F is applied on $g_c(t)$ to match the degraded image $f_c(t\text{-}k)$. $\eta(t,k)$ is the additive Gaussian noise with autocorrelation matrix $W^1(t,k)$. All these matrices are assumed to be known in [33]. Then, Least Square error is defined as

$$\varepsilon^2(t) = \sum_{k=0}^{\infty} \left\| f_c(t-k) - DH(t-k)F(t,k)g_c(t) \right\|^2_{W(t,k)} \tag{2.65}$$

Ideal image sequence minimizing above error term can be found using iterative techniques. Applying LMS algorithm [32] to the problem

$$\hat{g}(t) = F^{\perp}(t,1)\hat{g}(t-1) - \frac{\mu}{2} \left. \frac{\partial \varepsilon^2(t)}{\partial \hat{g}(t)} \right|_{F^{\perp}(t,1)\hat{g}(t-1)} \tag{2.66}$$

where $F^{\perp}(t,1)$ is the forward geometric warp operator. $F^{\perp}(t,1)$ is the pseudo-inverse of $F(t,1)$. The estimate of the ideal image sequence is updated at each iteration using steepest descent rule.

2.9. Reconstruction via Alias Removal

In this method [38], an HR image is reconstructed using the shifts (translational offsets on a subpixel level) between LR images. Only global motion is assumed between the images. A gradient based shift estimator is used for image registration purpose.

The gradient method is based on Taylor series expansion. Using Taylor series expansion, a smooth function *f(x)* can be estimated at point *x* as (deleting the higher order terms)

$$f(x) \cong f(a) + (x-a)f'(a) \tag{2.67}$$

where *a* is close to *x*. If *f(x)*, *f(a)* and *f'(a)* are known, then *(x-a)* can be found. Using 2D discrete coordinates for digital images

$$f(m,n) \cong f(m_0,n_0) + (m-m_0)\frac{\partial f(m_0,n_0)}{\partial m} + (n-n_0)\frac{\partial f(m_0,n_0)}{\partial n} \tag{2.68}$$

$(m-m_0)$ and $(n-n_0)$ represents the shifts. These shifts are denoted as t_x and t_y. After calculating the gradients for each pixel, the method of least squares can be used to solve for the shifts. Least squares error can be expressed as

$$Error = \frac{1}{MN}\sum_{m}^{M}\sum_{n}^{N}\left[f(m,n) - f(m_0,n_0) - t_x\frac{\partial f(m_0,n_0)}{\partial m} - t_y\frac{\partial f(m_0,n_0)}{\partial n}\right]^2 \tag{2.69}$$

where *M* and *N* are the total number of samples in *x* and *y* directions respectively. Taking the gradient of the error expression and equating the result to zero, we get

$$\begin{bmatrix} \sum_{m}^{M}\sum_{n}^{N}\left(\frac{\partial f(m_0,n_0)}{\partial m}\right)^2 & \sum_{m}^{M}\sum_{n}^{N}\frac{\partial f(m_0,n_0)}{\partial m}\frac{\partial f(m_0,n_0)}{\partial n} \\ \sum_{m}^{M}\sum_{n}^{N}\frac{\partial f(m_0,n_0)}{\partial m}\frac{\partial f(m_0,n_0)}{\partial n} & \sum_{m}^{M}\sum_{n}^{N}\left(\frac{\partial f(m_0,n_0)}{\partial n}\right)^2 \end{bmatrix} * \begin{bmatrix} t_x \\ t_y \end{bmatrix} = \begin{bmatrix} \sum_{m}^{M}\sum_{n}^{N}\left(f(m,n)-f(m_0,n_0)\right)\frac{\partial f(m_0,n_0)}{\partial m} \\ \sum_{m}^{M}\sum_{n}^{N}\left(f(m,n)-f(m_0,n_0)\right)\frac{\partial f(m_0,n_0)}{\partial n} \end{bmatrix} \tag{2.70}$$

From this system of equations, the shifts can be found easily. The gradients can be estimated using various methods such as Sobel and Prewitt. Also, using a low pass filter before gradient calculation can significantly improve the results of shift estimator.

After shift estimation, an alias free spectrum can be estimated from a set of aliased spectra.

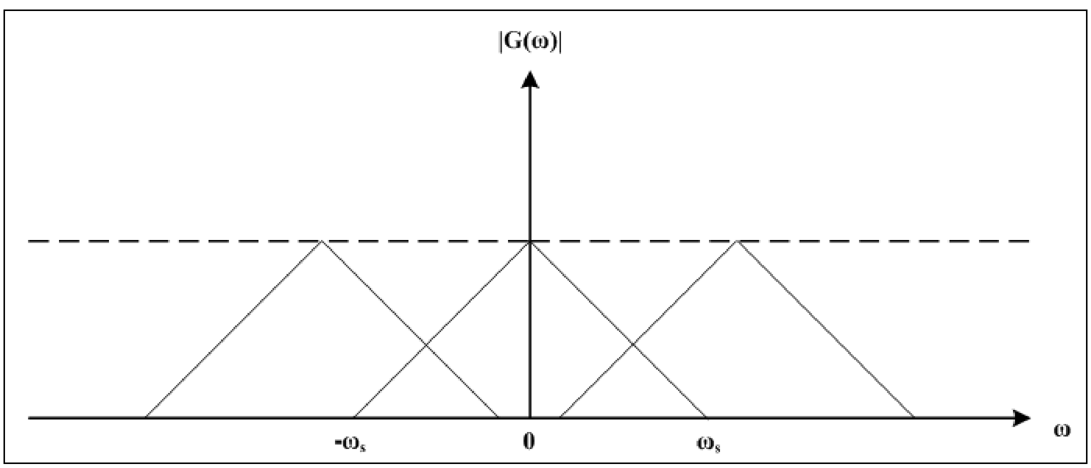


Figure 2.12. Aliased spectrum as a summation of unaliased spectra

In Figure 2.12, dashed line represents aliased spectrum and the triangles represents the spectrum of the original HR signal. The frequency unit is radians per second. Repeated spectra occur as the result of sampling of continuous signal. According to [38], if we know the bandlimit of the HR image, then unaliased spectra of the HR signal can be recovered using multiple signals (images in our case).

The reconstruction method is based on the shifting property of Fourier transform. Let $G_c(u,v)$ denote the CFT of the HR version of the reference image. Then the CFT of temporally neighboring images are related to CFT of the reference image through the relation below:

$$G_{ck}(u,v) = e^{j2\pi(t_x u + t_y v)} G_c(u,v) \tag{2.71}$$

Here $G_{ck}(u,v)$ is the CFT of the kth image, u and v are the continuous frequency variables and t_x, t_y are the shift variables along x and y directions respectively. Motion between images is modeled as global motion. From the well known shift property of Fourier transform, shift in time domain corresponds to multiplication with a complex exponential in frequency domain.

Since, we are dealing with digital images, above equation should be discretized. DFT simply means the sampled version of DTFT and DTFT is the sum of the repeated aliases of CFT. Using this fact

$$F_k(m,n) = \frac{1}{T_x T_y} \sum_{r=-\infty}^{\infty} \sum_{c=-\infty}^{\infty} G_{ck}\left(\frac{2\pi m}{MT_x} + r\omega_x, \frac{2\pi n}{NT_y} + c\omega_y \right) \tag{2.72}$$

for $m = 0,1 \ldots M-1$ and $n = 0,1 \ldots N-1$. Here $F_k(m,n)$ is the DFT of the kth LR image (note the discrete indices). Also, T_x and T_y are the sampling periods and ω_x and ω_y are sampling frequencies in x and y directions respectively.

As stated before, multiple images are used to find the unaliased spectra. To find the number of images, the criteria is

$$|G_c(u,v)| = 0 \qquad \text{for} \qquad |u| > L_x \omega_x \quad \text{and} \quad |v| > L_y \omega_y \tag{2.73}$$

This means one input LR image is used for each overlapping spectra. Then, infinite summations in the equation reduces to

$$F_k(m,n) = \frac{1}{T_x T_y} \sum_{r=-L_x}^{L_x-1} \sum_{c=-L_y}^{L_y-1} G_{ck}\left(\frac{2\pi m}{MT_x} + r\omega_x, \frac{2\pi n}{NT_y} + c\omega_y\right) \tag{2.74}$$

Note that this equation can also be written in matrix form.

$$\mathrm{F}_{\mathrm{mn}} = \varphi_{\mathrm{mn}} \mathrm{G}_{\mathrm{mn}} \tag{2.75}$$

In this equation, F_{mn} contains the DFT values of each LR image at discrete frequency point *(m,n)*. The matrix φ_{mn} contains the phase shift information (due to global motion) for each signal. Finally, G_{mn} includes the DFT values of HR reference image (samples of $G_{ck}(u,v)$). Finding G_{mn} for each *(m,n)* location, we get the DFT of HR image. Taking inverse DFT, we can find unalised HR image.

Note that, φ_{mn} has a dimension of $px4L_xL_y$. p is the number of LR frames at hand and $4L_xL_y$ is the number of samples taken from $G_{ck}(u,v)$. In the ideal case, $p=4L_xL_y$ and φ_{mn} is a square matrix. So, solution can be found by simply matrix inversion. If φ_{mn} is not square, then a pseudo-inverse or least square solution can be obtained.

2.10.Recursive Least Squares

In [39] and [40], the idea is the same with [38]. But, Kim and Su also included the effect of additive noise and blur. Figure 2.13 shows the overall structure of the problem of HR image reconstruction from blurred noisy LR images.

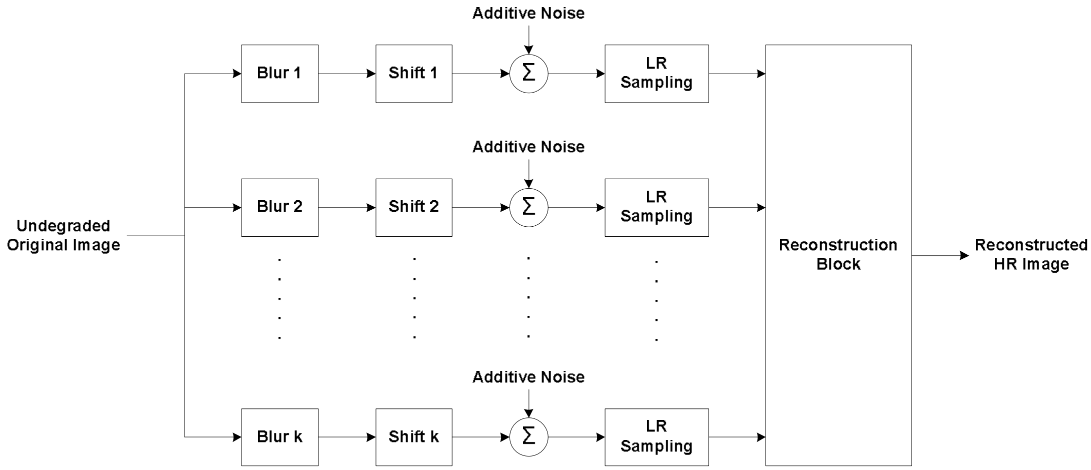


Figure 2.13. Block diagram of HR reconstruction system.

As the first step, blur is applied to undegraded original image. Let $G_c(u,v)$ denote the CFT of the HR version (undegraded) of the image. Then the CFT of temporally neighboring images are related to CFT of the undegraded image through the relation below:

$$G_{ck}(u,v) = H_k(u,v)e^{j2\pi(t_x u + t_y v)}G_c(u,v) \tag{2.76}$$

Here $G_{ck}(u,v)$ is the CFT of the kth image, u and v are the frequency variables and t_x, t_y are the shift variables along x and y directions respectively. $H_k(u,v)$ is the blur operator for kth image. Then, motion between LR images is modeled as global motion. Time shift corresponds to multiplication with a complex exponential in frequency domain. The DFT of kth LR image can be written as

$$F_k(m,n) = \frac{1}{T_x T_y}\sum_{r=-L_x}^{L_x-1}\sum_{c=-L_y}^{L_y-1} G_{ck}\left(\frac{2\pi m}{MT_x} + r\omega_x, \frac{2\pi n}{NT_y} + c\omega_y\right) \tag{2.77}$$

for $m = 0,1 \ldots M-1$ and $n = 0,1 \ldots N-1$. Here $F_k(m,n)$ is the DFT of the kth LR image (note the discrete indices). Also, T_x and T_y are the sampling periods and ω_x and ω_y are sampling frequencies in x and y directions respectively. L_x and L_y are found from

$$\left|G_c(u,v)\right| = 0 \qquad \text{for} \qquad |u| > L_x\omega_x \quad \text{and} \quad |v| > L_y\omega_y \tag{2.78}$$

In matrix form,

$$F_{mn} = \varphi_{mn} G_{mn} \tag{2.79}$$

Including the effect of additive noise, the equation can be rewritten as

$$Z_{mn} = \varphi_{mn} G_{mn} + N_{mn} \tag{2.80}$$

In this equation, Z_{mn} contains the DFT values of each LR image at discrete frequency point *(m,n)*. The matrix φ_{mn} contains the phase shift information (due to global motion) and effect of blur for each signal. N_{mn} contains the DFT values of the noise components. Finally, G_{mn} includes the DFT values of HR reference image (samples of $G_{ck}(u,v)$). Finding G_{mn} for each *(m,n)* location, we get the DFT of HR image. Taking inverse DFT, we can find unalised HR image.

If $A=\varphi_{mn}$, $g=G_{mn}$, $b=Z_{mn}$, then equation 2.80 can be written as $Ag=b$. A is known to be highly ill-conditioned [40]. So the pseudo-inverse solution, $g=(A^TA)^{-1}A^Tb$, is not accurate enough. For HR reconstruction, a regularized recursive total least squares method can be used. Tikhonov regularization is incorporated to obtain a stable and more accurate solution. In [39] and [40], the proposed objective function is

$$R(g) = \|Ag - b\|^2 + \lambda \|g - c\|^2 \tag{2.81}$$

The solution minimizing $R(x)$ is

$$g = \left(A^T A + \lambda I\right)^{-1} \left(A^T b + \lambda c\right) \tag{2.82}$$

where λ is a regularization parameter. At first estimation $c=0$. If number of frames p is greater than $4L_xL_y$, the regularized least square solution for $(p+1)$*-th* estimate is obtained by setting $c(p+1)=G_{mn}(p)$. So iterative reconstruction formula is

$$G_{mn}(p+1) = \left[\varphi_{mn}^T(p+1)\varphi_{mn}(p+1) + \lambda I\right]^{-1} \left[\varphi_{mn}^T(p+1) Z_{mn}(p+1) + \lambda G_{mn}(p)\right] \tag{2.83}$$

3. IMAGE INTERPOLATION METHODS

In this chapter, seven interpolation methods are covered. These methods are cubic convolution, cubic B-spline interpolation, Lanczos interpolation, wavelet based image interpolation, edge adaptive interpolation, interpolation using MRF and interpolation using exponential based interpolation functions.

3.1. Cubic Convolution

If there is a sampled function $f(m,n)$, then its continuous counterpart $f_c(x,y)$ can be reconstructed using a suitable interpolation kernel. In [41] and [42], 1D kernel for cubic convolution is derived. 2D interpolation can be accomplished applying 1D interpolation on horizontal and vertical directions successively. For equally spaced data, $f_c(x,y)$ can be written as

$$f_c(x,y) = \sum_m \sum_n f(m,n)\phi(x-m)\phi(y-n) \tag{3.1}$$

In this expression, $\phi()$ is the 1D interpolation kernel, (m,n)'s are the interpolation nodes and $f(m,n)$ is the value of f at these nodes. In [41], the cubic convolution kernel is given as:

$$\phi(x) = \begin{cases} 1.5|x|^3 - 2.5|x|^2 + 1 & 0 \le |x| < 1 \\ -0.5|x|^3 + 2.5|x|^2 - 4|x| + 2 & 1 \le |x| < 2 \\ 0 & 2 \le |x| \end{cases} \tag{3.2}$$

After finding $f_c(x,y)$, this continuous image is resampled again at the desired resolution to obtain the HR image (refer to Figure 3.1).

Figure 3.1. Interpolating an HR image from an LR image

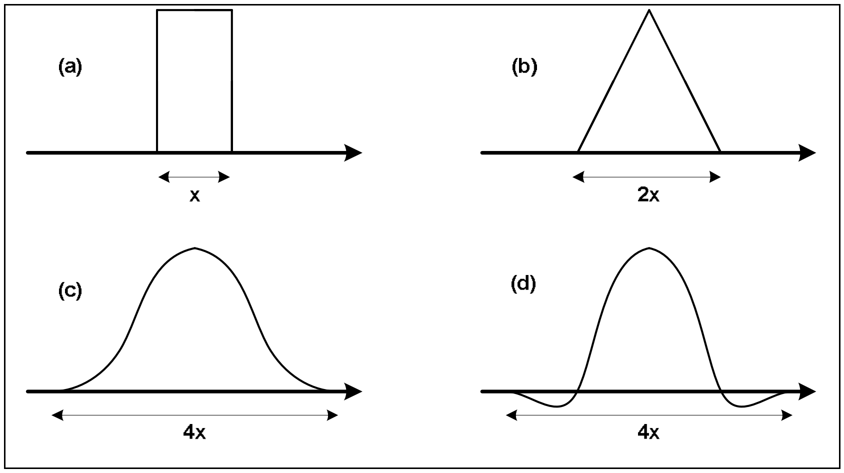


Figure 3.2. Different interpolation kernels (a) Nearest neighbor (b) Linear interpolation (c) Cubic B-spline (d) Cubic convolution (eq. 3.2)

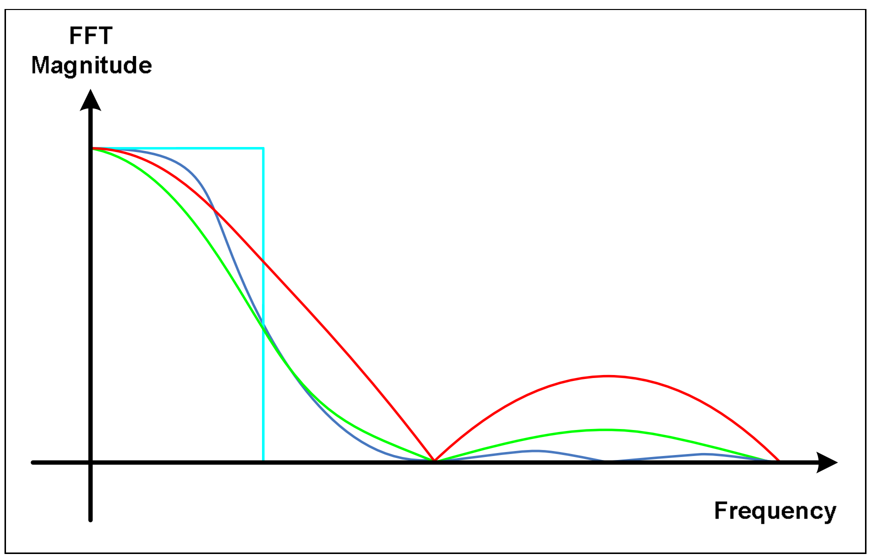


Figure 3.3. Amplitude spectra of interpolation kernels, red for nearest neighbor, blue for cubic and green for linear interpolation kernels

As can be seen from Figure 3.3, the spectrum of cubic convolution is closer to the spectrum of ideal filter. Here red, blue and green curves are the FFT magnitudes of nearest neighbor, cubic and linear interpolation kernels, respectively.

3.2. Cubic B-Spline Interpolation

B-splines can also be a good choice for image interpolation [43]. First and second order B-splines corresponds to nearest neighbor and linear interpolation kernels.

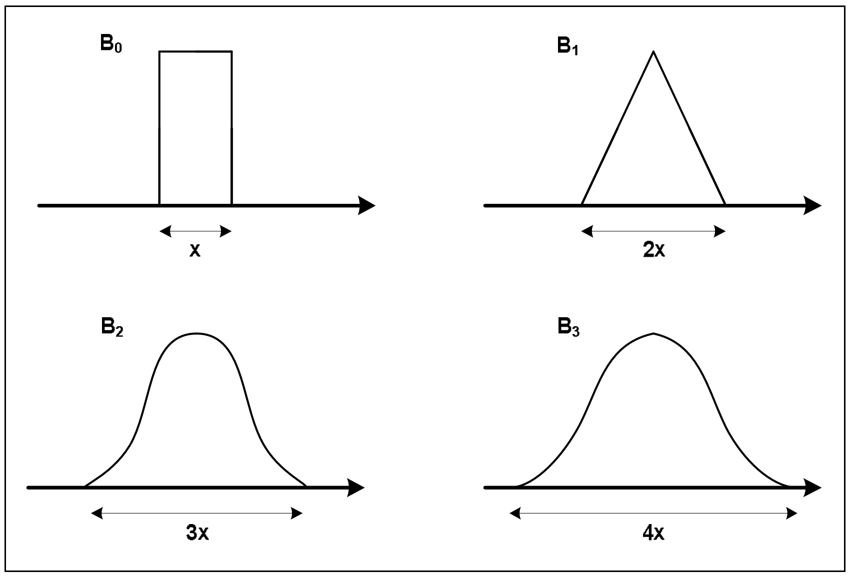


Figure 3.4. *B*-spline functions

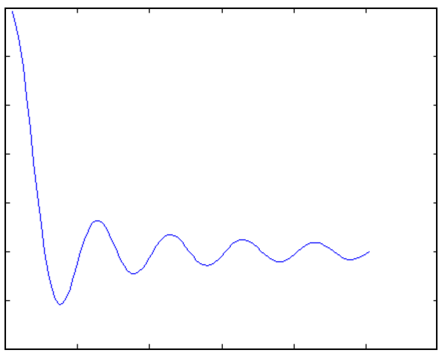
Cubic B-spline kernel is depicted in Figure 3.4 and is given as

$$B_3(x) = \begin{cases} \left(3|x|^3 - 6|x|^2 + 4\right)/6 & 0 \le |x| < 1 \\ (2-|x|)^3/6 & 1 \le |x| < 2 \\ 0 & 2 \le |x| \end{cases} \tag{3.3}$$

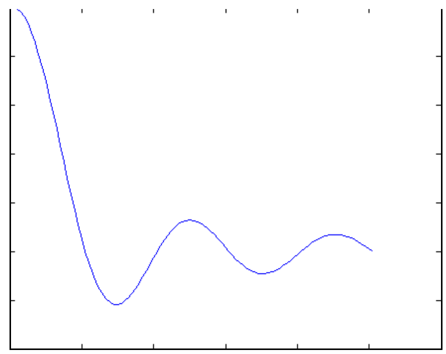
3.3. Lanczos Interpolation

Lanczos interpolation is a frequently used interpolation method in graphics applications. Lanczos kernel is obtained by windowing the sinc function with another sinc function. The kernel shown in Figure 3.5-c is obtained by windowing the sinc in Figure 3.5-a with the sinc function in Figure 3.5-b. The two lobed kernel is defined in [44] as

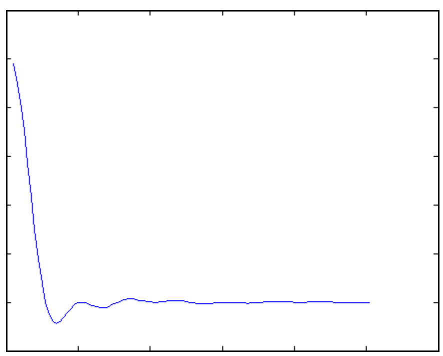
$$L(x)=\begin{cases}\dfrac{\sin(\pi x)}{\pi x}\dfrac{\sin\left(\pi\dfrac{x}{2}\right)}{\pi\dfrac{x}{2}} & |x|<2\\ 0 & |x|\geq 2\end{cases} \tag{3.4}$$



(a)



(b)



(c)

Figure 3.5. (a) *sinc(x)*, (b) *sinc(x/2)*, (c) *sinc(x)sinc(x/2)*

3.4. Wavelet Based Image Interpolation

In [47], Temizel and Vlachos discussed image interpolation problem in wavelet transform domain. They assumed that LR image at hand is the lowpass filtered and decimated subband of the HR image. A trivial approach is to fill the HR subbands with zero value and taking inverse wavelet transform. A better method is cyclespinning. Since decimated wavelet transform is shift-invariant, quantization of the coefficients or inexact computation of high frequency coefficients cause ringing and similar artifacts around the discontinuities at the resultant image. In [47], it is said that cyclespinning is very useful in suppressing these artifacts.

First, an HR estimate g is generated by assigning zero value to the unknown high frequency subbands and taking inverse wavelet transform. Then, a number of LR images f_i are generated from g by applying shift to g in different directions and discarding high frequency components in wavelet domain. Zero value is assigned to high frequency components of f_i images and inverse wavelet transform is applied to get corresponding high resolution g_i images. Final result is obtained by aligning and averaging g_i images.

In [48], Temizel and Vlachos proposed another wavelet based image interpolation method. As denoted above, one of the most useful properties of wavelet transform is persistency which means, wavelet coefficients tend to propagate from lower resolution to higher resolution scales. However, this property is not valid for coefficient signs, [48].

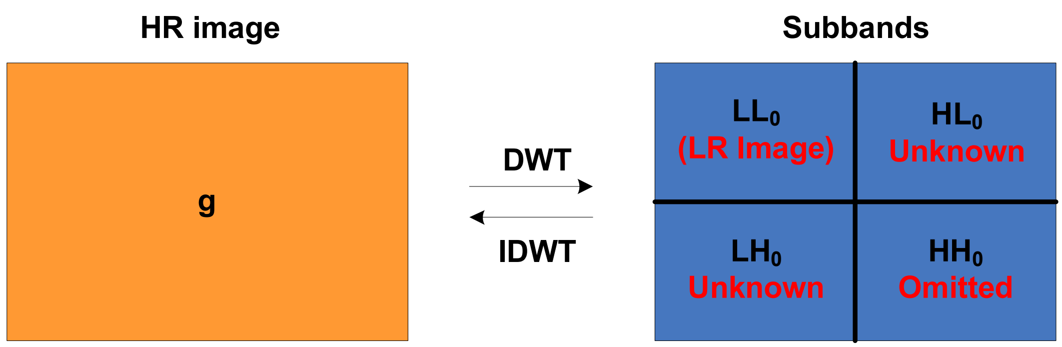


Figure 3.6. Original image and its sub-bands

$g(z_1,z_2)$, $L(z_1)$ and $H(z_1)$ denote the HR image we want to estimate, high pass filter and low pass filter, respectively (refer to Figure 3.6).

$$LL_0(z_1,z_2) = (\downarrow 2)L_{col}(z_2)L_{row}(z_1)g(z_1,z_2) \tag{3.5}$$

and

$$HL_0(z_1,z_2) = (\downarrow 2)L_{col}(z_2)H_{row}(z_1)g(z_1,z_2) \tag{3.6}$$

where LL_0 and HL_0 are the low and high frequency subbands of g. $(\downarrow 2)$ denotes decimation by a factor of 2. Using this framework, LR input image is the LL_0 subband of the HR image g we want to estimate. Then, we can approximate the high frequency subband of g as

$$HL_0'(z_1, z_2) = H_{row}(z_1)LL_0(z_1, z_2) \tag{3.7}$$

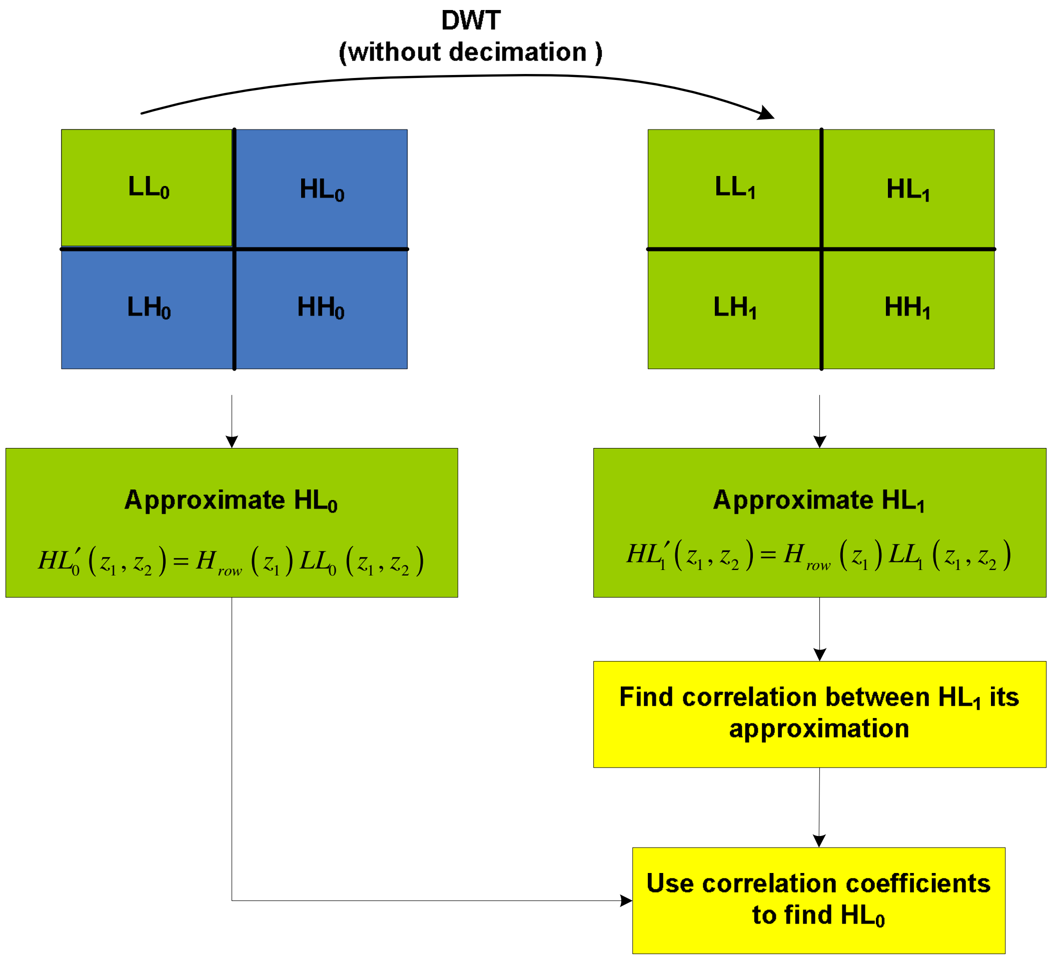


Figure 3.7. Image interpolation in wavelet domain

HL_0 will be found using HL_0' and the correlation coefficients between HL_0 and HL_0'. To find the correlation between HL_0 and HL_0', we go one scale further in wavelet domain (refer to Figure 3.7)

$$LL_1(z_1, z_2) = L_{col}(z_2)L_{row}(z_1)LL_0(z_1, z_2) \tag{3.8}$$

$$HL_1(z_1, z_2) = L_{col}(z_2)H_{row}(z_1)LL_0(z_1, z_2) \tag{3.9}$$

and

$$HL_1'(z_1, z_2) = H_{row}(z_1) LL_1(z_1, z_2) \tag{3.10}$$

The correlation coefficients between the components of HL_1 and HL_1' are computed using linear least squares estimation. The wavelet coefficient at *(m,n)* position of HL_1 and the wavelet coefficients at *(m-1,n)*, *(m,n)*, *(m+1,n)* and *(m+2,n)* positions of HL_1' are used in estimation process. As can be understood from the coefficients above, horizontal correlation is taken into account while estimating vertical subband component (vertical correlation is taken into account while estimating horizontal subband component). If we denote the wavelet coefficient at *(m,n)* position of HL_1 as $y_{m,n}$ and the wavelet coefficients at *(m-1,n)*, *(m,n)*, *(m+1,n)* and *(m+2,n)* positions of HL_1' as $x_{m-1,n}$, $x_{m,n}$, $x_{m+1,n}$ and $x_{m+2,n}$

$$y_{m,n} = c_0 + c_1 x_{m-1,n} + c_2 x_{m,n} + c_3 x_{m+1,n} + c_4 x_{m+2,n} \tag{3.11}$$

If we write this equation for every pixel in the neighborhood of *(m,n)* and solve the system of equations, we get correlation coefficients c_0, c_1, c_2, c_3 and c_4. These correlation coefficients are then used to estimate HL_0 using HL_0'. Same procedure is repeated for LH_0. HH_0 subbband is omitted in [48]. Finally, inverse wavelet transform is applied and the result is the interpolated HR image.

3.5. Edge Adaptive Image Interpolation

In [49], Mori, Kameyama, Ohmiya, Lee and Toraichi proposed an edge-adaptive interpolation method where interpolation kernel rotates according to the direction of the gradient. For an image of *MxN* pixels, the interpolation formula is given as

$$f_c(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) \psi(x-m, y-n) \tag{3.12}$$

In this expression, *(m,n)*'s are the interpolation nodes, *f(m,n)* is the value of *f* at these nodes and $\psi(x, y)$ is 2D interpolation kernel which can be bicubic, bilinear and has a finite support.

$$\psi(x, y) = 0 \quad \text{for} \quad |x| > c \quad \text{or} \quad |y| > c \tag{3.13}$$

Rewriting the interpolation formula

$$f_c(x,y)=\sum_{m=-(c-1)}^{c}\sum_{n=-(c-1)}^{c} f(\lfloor x\rfloor+m,\lfloor y\rfloor+n)\psi((x-\lfloor x\rfloor)-m,(y-\lfloor y\rfloor)-n) \quad (3.14)$$

Note that interpolation kernel can be implemented in a separable manner which is the cause of jaggy artifacts around edges.

$$\psi(x,y)=\phi(x)\phi(y) \quad (3.15)$$

In [49], the proposed kernel is given as (refer to Figure 3.8)

$$\psi'(x,y)=\begin{cases}\phi(x)\phi(y-kx) & |k|\le 1\\ \phi\left(x-\dfrac{y}{k}\right)\phi(y) & |k|>1\end{cases} \quad (3.16)$$

where

$$k=h_y/h_x \qquad g=grad(u) \qquad h\perp g \qquad h=(h_x,h_y) \quad (3.17)$$

And new interpolation formula is

$$f_c(x,y,k)=\begin{cases}\displaystyle\sum_{m=-(c-1)}^{c}\sum_{n=-(c-1)}^{c} f(\lfloor x\rfloor+m,\lfloor y\rfloor+n+\lfloor l\rfloor)\phi((x-\lfloor x\rfloor)-m)\phi(l_f-n) & |k|\le 1\\ \displaystyle\sum_{m=-(c-1)}^{c}\sum_{n=-(c-1)}^{c} f(\lfloor x\rfloor+m+\lfloor l\rfloor,\lfloor y\rfloor+n)\phi(l_f-m)\phi((y-\lfloor y\rfloor)-n) & |k|>1\end{cases}$$

(3.18)

$$l=\begin{cases}k(i-(x-\lfloor x\rfloor))+(y-\lfloor y\rfloor) & |k|\le 1\\ k(j-(y-\lfloor y\rfloor))+(x-\lfloor x\rfloor) & |k|>1\end{cases} \qquad \text{and} \qquad l_f=l-\lfloor l\rfloor \quad (3.19)$$

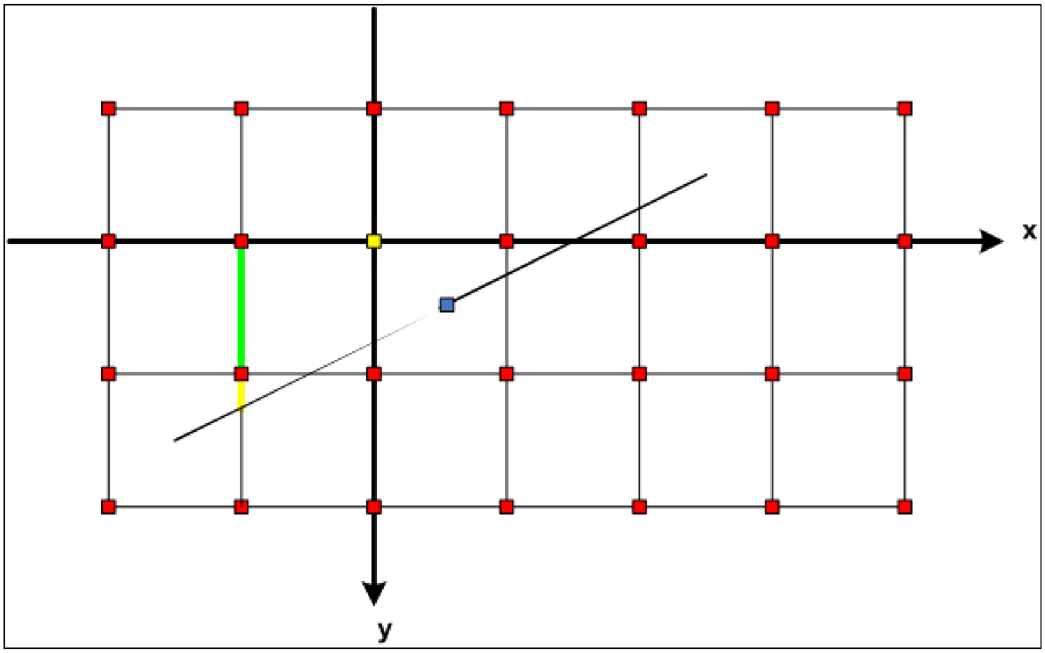


Figure 3.8. Edge with k<1

In Figure 3.8, blue square (x_0, y_0) is the point to be calculated. The equation of the line passing over blue square $y = k(x - x_0) + y_0$. Yellow square is $(\lfloor x_0 \rfloor, \lfloor y_0 \rfloor)$. The length of the green line is $\lfloor l \rfloor$. And the length of the yellow line is l_f.

In [50], Carrato, Ramponi and Marsi assumed that imaging process has two stages as low-pass filtering and decimation. The low-pass filtering operation modifies the values of the pixels near the edges proportional to the distance between pixels and the edge. So, an analysis of the values of the low resolution pixels gives an idea about the sub-pixel position of the edge. One dimensional case is illustrated in Figure 3.9.

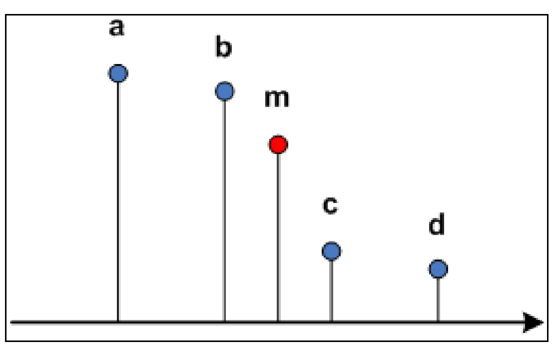


Figure 3.9. One possible position for the edge to be reconstructed: a, b, c, d are LR pixels and m is HR pixel

Using the proposed method [50],

$$m = \mu b + (1-\mu)c \tag{3.20}$$

$$\mu = \frac{k(c-d)^2 + 1}{k\left((a-b)^2 + (c-d)^2\right) + 2} \tag{3.21}$$

k is an input parameter. If k=0, then method works as linear interpolation. As k increases, edge sensitivity increases. When edge is in midway between b and c, a-b=c-d and $m=(b+c)/2$. When edge is closer to c, then a-b<c-d and m takes a value closer to b.

Two dimensional case is illustrated on Figure 3.10. First P_1 and P_2 pixels are interpolated using 1D interpolation kernels in the horizontal (red) and vertical (blue) directions respectively. Then pixel P_3 is calculated using interpolation kernels in the horizontal and vertical directions separately. The mean value of the two results is assigned as the final value of P_3 pixel.

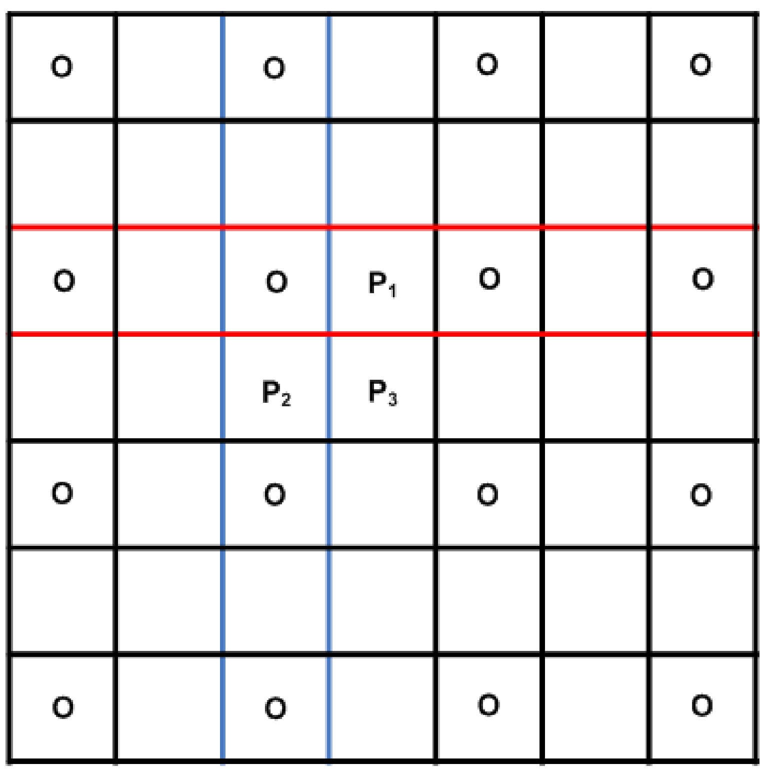


Figure 3.10. Two dimensional case in edge adaptive interpolation

3.6. Image Interpolation Using Wide Sense Markov Random Fields

In [51], Nemirovsky and Porat proposed a new texture interpolation method based on wide sense Markov random fields. Markov random fields are well-suited for representing the spatial correlation of most images. In this model, image texture is composed of two orthogonal components: a purely indeterministic component and a deterministic component. The purely indeterministic component is modeled as a two dimensional auto-regressive process and the deterministic component is expressed as a sum of sinusoids.

To find the deterministic part (refer to Figure 3.11), first, DFT and periodogram of the LR image is computed. The peaks of the periodogram are found and a frequency domain filter is generated whose value is 1 at the peaks and 0 at other points. DFT is filtered with this mask. Zero padding the filtered DFT and applying inverse DFT we get deterministic component of HR image.

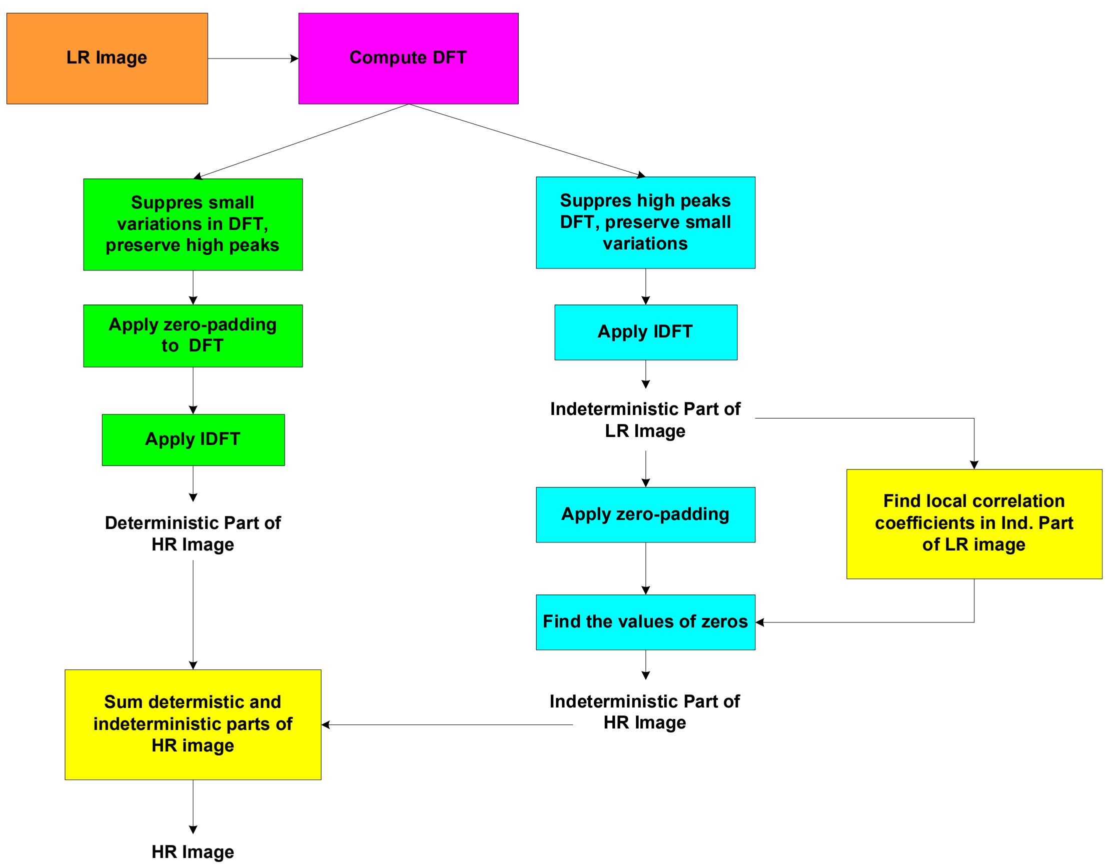


Figure 3.11. Computation of deterministic and indeterministic parts of HR image

To obtain the indeterministic part of the LR image (refer to Figure 3.11), the DFT of LR image is filtered with the negative version of the filter described above. Indeterministic part of the LR image is computed by applying inverse DFT to the filtered DFT. After zero padding the indeterministic part of the LR image, the holes are filled as described below.

For a discrete random field *f*, the linear least square estimate of *f(m,n)* can be found as the linear combination of the values in the causal neighborhood of *(m,n)*.

$$\hat{f}(m,n) = \sum_{\substack{i,j \\ (m-i,n-j)\in X_{m,n}}} c_{i,j} f(m-i,n-j) \tag{3.22}$$

where the coefficients are found by minimizing the following error.

$$e_{m,n} = E\left\{\left[f(m,n) - \hat{f}(m,n)\right]^2\right\} \tag{3.23}$$

In [51], 3 nearest neighbors are used.

$$D = \{(0,1),(1,1),(1,0)\} \tag{3.24}$$

$$\hat{f}(m,n) = \sum_{(i,j)\in D} c_{i,j} f(m-i,n-j) \tag{3.25}$$

There remain 3 equations for three unknown coefficients after some derivation.

$$c_{1,0}R_{ff}(0,0) + c_{1,1}R_{ff}(0,1) + c_{0,1}R_{ff}(-1,1) = R_{ff}(-1,0) \tag{3.26}$$

$$c_{1,0}R_{ff}(0,1) + c_{1,1}R_{ff}(0,0) + c_{0,1}R_{ff}(-1,0) = R_{ff}(-1,1) \tag{3.27}$$

$$c_{1,0}R_{ff}(1,-1) + c_{1,1}R_{ff}(1,0) + c_{0,1}R_{ff}(0,0) = R_{ff}(0,-1) \tag{3.28}$$

where

$$R_{ff}(\alpha,\beta) = E\{f(m,n)f(m+\alpha,n+\beta)\} \tag{3.29}$$

is the autocorrelation matrix of the random field *f(m,n)*. These three coefficients give the relation between the pixels of LR image. The relation between the pixels of HR image is obtained from these coefficients as described in [51]. After finding deterministic and indeterministic parts of HR image, these two are summed to find the final HR image.

3.7. Exponential Based Interpolation Functions

In [52], Kirshner and Porat discussed interpolation problem in Sobolev space framework. Sobolev spaces consist of smooth functions and they can be used to approximate a continuous finite energy signal. The sampling process is interpreted as an orthogonal projection. In [52], third order reproducing kernel $\varphi(s,t)$ is used for interpolation purpose.

$$\varphi(s,t) = \frac{1}{16} e^{-|s-t|} \left(3 + 3|s-t| + |s-t|^2\right) \tag{3.30}$$

Let *P* be the orthogonal projection (sampling operator)

$$Px(\tau) = cG^{-1}b \tag{3.31}$$

where τ is the interpolation coordinate, $Px(\tau)$ is the interpolated value, *c* vector consists of *N* neighboring pixels in the neighborhood of τ, *b* vector has a size of *N* and consists of the samples of reproducing kernel

$$b_n = \varphi(\tau, t_n) \tag{3.32}$$

where t_n is a sample coordinate and $n = 1, \ldots, N$. *G* is NxN Gram matrix and is given by

$$G(m,n) = \varphi(t_m, t_n) \tag{3.33}$$

In [52], *N* is 11. *c* vector is depicted in Figure 3.12. Yellow pixel is at τ coordinate, red pixels are the neighbor pixels. For 2D signal, seperability property can be used. Interpolation is accomplished in two stages, first row wise than column wise.

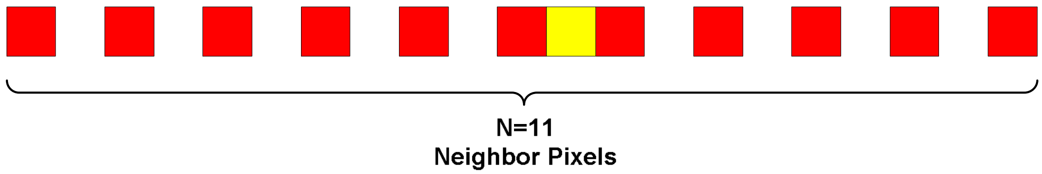


Figure 3.12. The interpolation coordinate (yellow one) and neighbor pixels (red ones)

4. IMAGE AND VIDEO QUALITY ASSESMENT METHODS

4.1. Peak Signal to Noise Ratio (PSNR)

PSNR is computed by dividing the possible peak power of the signal by *MSE*. Let x and y be the original and distorted images respectively.

$$MSE = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\left\| x(m,n) - y(m,n)\right\|^2 \tag{4.1}$$

Here M and N are the height and width of the images.

$$PSNR = 10\log_{10}\left(\frac{\left(2^W - 1\right)^2}{MSE}\right) \tag{4.2}$$

W is the number of bits to represent each pixel (for a gray level image). *PSNR* is expressed in log. dB scale.

PSNR is largely used in image and video processing literature because of its mathematical tractability. However, *PSNR* generally does not correlate with the preferences of the human visual system, [56]. In [53], it is said that *PSNR* is a reliable metric in video coding applications.

4.2. Universal Image Quality Index

In [54], Wang and Bovik proposed a metric that models the combination of three types of distortion, loss of correlation, luminance distortion and contrast distortion. Let x and y be the original and corrupted images respectively. Then, quality metric, Q is

$$Q = \frac{4\sigma_{xy}\overline{x}\,\overline{y}}{\left(\sigma_x^2 + \sigma_y^2\right)\left[\left(\overline{x}\right)^2 + \left(\overline{y}\right)^2\right]} \tag{4.3}$$

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad \bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i \tag{4.4}$$

$$\sigma_x^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2 \qquad \sigma_y^2 = \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \bar{y})^2 \tag{4.5}$$

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y}) \tag{4.6}$$

where N is the number of samples in each image. Q can be written as a product of three components

$$Q = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \frac{2\bar{x}\,\bar{y}}{(\bar{x})^2 + (\bar{y})^2} \frac{2\sigma_x \sigma_y}{\sigma_x^2 + \sigma_y^2} \tag{4.7}$$

The first component is the correlation coefficient between x and y, the second component measures how close the mean luminance values of x and y, and third component measures the similarity of the contrast values of x and y.

For image quality assessment, this metric is computed locally for each pixel using a local neighborhood. Then, Q metric for the image is obtained by averaging the values calculated for each pixel.

$$Q = \frac{1}{N}\sum_{i=1}^{N} Q_i \tag{4.8}$$

Q_i is the value for ith pixel and Q is the value for the whole image.

4.3. Single Scale Structural Similarity

In [55], a structural similarity measure is introduced to model the structural adaptivity of human visual system. Let x and y be two images we want to compare. μ_x is the mean of x and is used as an approximation to the luminance of x. σ_x^2 is the variance of x and is used as an approximation to contrast of x. σ_{xy} is the covariance of x and y and is a measure of the structural similarity between x and y. In [55], the luminance, contrast and structural comparison measures are given as

$$l(x,y)=\frac{2\mu_x\mu_y+C_1}{\mu_x^2+\mu_y^2+C_1} \tag{4.9}$$

$$c(x,y)=\frac{2\sigma_x\sigma_y+C_2}{\sigma_x^2+\sigma_y^2+C_2} \tag{4.10}$$

$$s(x,y)=\frac{\sigma_{xy}+C_3}{\sigma_x\sigma_y+C_3} \tag{4.11}$$

where

$$C_1=(0.01L)^2 \qquad C_2=(0.03L)^2 \qquad C_3=C_2/2 \tag{4.12}$$

L is the number of bits to represent a pixel in a gray level image. Finally, single scale structural similarity measure is given as

$$SSIM(x,y)=[l(x,y)]^{\alpha}[c(x,y)]^{\beta}[s(x,y)]^{\gamma} \tag{4.13}$$

α, β and γ are used to adjust the relative importance of each component. If we take

$$\alpha=\beta=\gamma=1 \tag{4.14}$$

and

$$C_1=C_2=0 \tag{4.15}$$

then, *SSIM* is equivalent to universal image quality index given in [54]. C_1 and C_2 components on the denominators make *SSIM* more stable. As in universal image quality index, *SSIM* is also calculated locally and overall image quality value is found by averaging all local values.

4.4. Multi Scale Structural Similarity

Image perception depends on three factors, sampling density of the image signal, observer's visual system, the distance between the observer and image plane [56]. Single scale *SSIM* method defined above is only suitable for a specific setting. A multiscale *SSIM* method is proposed in [56] to remedy the inefficiency of single scale method.

In the multiscale method, the two input images *x* and *y* are lowpass filtered and decimated by 2 at each scale. Original images are at Scale *1* and highest scale is Scale *M*. The luminance comparison is computed at the last scale. The contrast and structure comparison is made at every scale. The overall expression for multiscale *SSIM* is given as

$$SSIM(x,y)=[l_M(x,y)]^{\alpha_M}\prod_{j=1}^{M}[c_j(x,y)]^{\beta_j}[s_j(x,y)]^{\gamma_j} \tag{4.16}$$

4.5. Structural Similarity for Video

The quality assessment of video is accomplished in three stages: local region level, frame level and sequence level, [57]. Block diagram of the process is given in Figure 4.1. First, the sequence is transformed into YCbCr color space. Local assessment is done as in the single scale SSIM case described before. But in this case, local quality metric is not computed for every pixel. Instead some sampling locations are determined and local quality metric is computed only in these locations using 8x8 square neighborhoods. Number of sampling locations is denoted as R_s. Local quality metric is computed on Y, Cb and Cr channels separately and then combined into a single metric.

$$SSIM_{ij}=W_Y SSIM_{ij}^{Y}+W_{Cb}SSIM_{ij}^{Cb}+W_{Cr}SSIM_{ij}^{Cr} \tag{4.17}$$

where $SSIM_{ij}^{Y}$ denotes the local quality metric of the jth sampling location on the ith frame. Also, W_Y is the weight of Y channel. In [57], these weights are taken as

$$W_Y=0.8 \qquad W_{Cb}=0.1 \qquad W_{Cr}=0.1 \tag{4.18}$$

Then, the local quality values are combined to get a frame level quality value.

$$Q_i = \frac{\sum_{j=1}^{R_s} \omega_{ij} SSIM_{ij}}{\sum_{j=1}^{R_s} \omega_{ij}} \tag{4.19}$$

where Q_i denotes the quality value for the ith frame and ω_{ij} is the weight for the jth window in the ith frame.

Finally, the sequence level quality value is calculated

$$Q = \frac{\sum_{i=1}^{F} W_i Q_i}{\sum_{i=1}^{F} W_i} \tag{4.20}$$

where F is the number of frames in the video and W_i is the weight for the ith frame. In [57], it is assumed that dark regions do not attract attention so they should be assigned smaller weights.

$$\omega_{ij} = \begin{cases} 0 & \mu_x \le 40 \\ (\mu_x - 40)/10 & 40 < \mu_x \le 50 \\ 1 & \mu_x > 50 \end{cases} \tag{4.21}$$

Here μ_x is the mean value of sampling window on Y channel and is used as an estimate of local luminance.

Also, it is denoted [57] that this algorithm does not perform well when large global motion occurs. Therefore, frame weight decreases as the amount of global motion increases. Block-based motion estimation is used to find motion vector for each sampling window. Then, motion level of the ith frame is estimated as

$$M_i = \frac{\left(\sum_{j=1}^{R_s} m_{ij}\right) / R_s}{K_M} \tag{4.22}$$

where m_{ij} is the motion vector length for the jth sampling window of the ith frame. K_M is a normalization constant and is used as 16 in [57]. The weighting of frame is computed using motion level as

$$W_i = \begin{cases} \sum_{j=1}^{R_s} \omega_{ij} & M_i \le 0.8 \\ ((1.2 - M_i)/0.4)\sum_{j=1}^{R_s} \omega_{ij} & 0.8 < M_i \le 1.2 \\ 0 & M_i > 1.2 \end{cases} \quad (4.23)$$

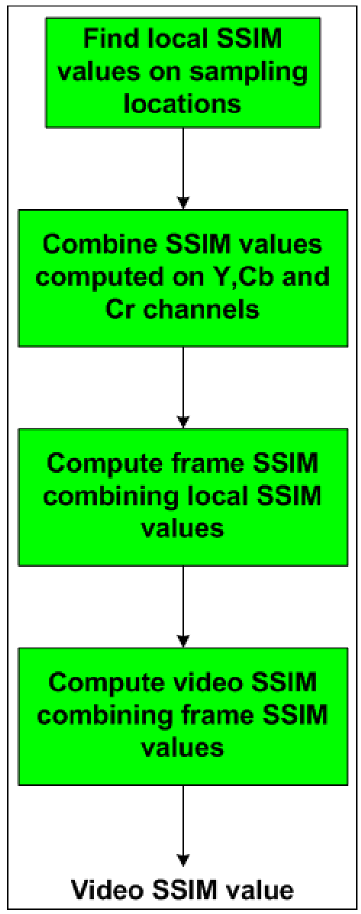


Figure 4.1. Computation of video SSIM value

5. EXPERIMENTAL RESULTS AND DISCUSSION

5.1. Objective Quality Assessment

PSNR and SSIM metrics were used to evaluate the results quantitatively. Three different video sequences: susie, table tennis and calendar sequences were used. These sequences were chosen according to their texture, smooth area and motion content. Each video sequence consists of 250 frames of 704x480 pixels. First step, the original sequences were decimated by a factor of two in the horizontal and vertical directions to obtain the LR video material. The decimated sequences consisted of 250 frames of 352x240 pixels. Then these three decimated sequences are resized to their original resolution (704x480) by seven different methods (refer to Figure 5.1), namely 1) robust super resolution, 2) LMS method, 3) bicubic interpolation, 4) wavelet based interpolation, 5) edge adaptive interpolation, 6) interpolation using wide sense Markov random fields and 7) interpolation using exponential based kernels as described in sections 2.7, 2.8, 3.1, 3.4, 3.5, 3.6 and 3.7, respectively.

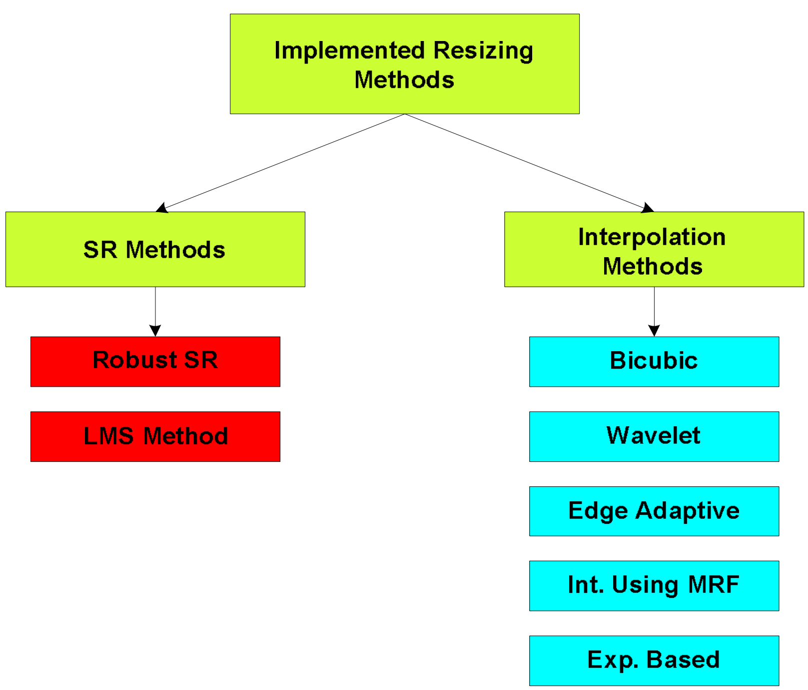


Figure 5.1. Implemented SR algorithms and interpolation methods

At this point, we have seven different results for each decimated video sequence. Then, PSNR and SSIM metrics were computed between processed sequences and original HR sequences. PSNR and SSIM values of each sequence for all methods are given in Table 5.1 and Table 5.2, respectively. PSNR value for each video sequence was found by averaging MSE values of all frames in sequence and then taking log of the average MSE and multiplying by 10. The first cell in the first row of Table 5.1 is the PSNR value of sequence 1 for LMS method and was found to be 34.6 dB. The outcome of this procedure for sequence 2 and sequence 3 take place in the 2nd and 3rd cells. The fourth cell in the first row of Table 5.1 is the average video PSNR value and is found by averaging MSE values of three video sequences and then taking log of the average MSE and multiplying by 10. The fifth cell in each row of Table 5.1 gives the rank of the corresponding method. The average value taking place in the fourth cell is used to determine their rank. Accordingly, the robust SR is the highest ranking method.

SSIM value for each sequence was found by using the method described in Section 4.5. The first cell in the first row of Table 5.2 is the SSIM value of sequence 1 for LMS method and was found to be 0.91. Similarly cells 2 and 3 contain the results for the sequence 2 and sequence 3. As SSIM value increases towards 1, the similarity between the original input and the processed output increases.

The fourth cell in the first row of Table 5.2 is the average of the SSIM values. The fifth cell in each row of Table 5.2 gives the rank of the respective method. The average value in the fourth cell is used to determine their rank. Robust SR is the most successful method also according to the SSIM measure. PSNR and SSIM values for each frame of three video sequences are given in Figure 5.2 and Figure 5.3, respectively.

Quantitative analysis was also conducted for frame stills. 100th and 200th frames were captured from each original sequence, as well as from the corresponding decimated-and-resized sequences of each method. 100th frames of the original video sequences are shown in Figure 5.4. PSNR and SSIM metrics between original and the resultant frame stills are given in Table 5.3 and Table 5.4, respectively. SSIM value for each frame was computed using the method described in Section 4.3.

Table 5.1. PSNR values of all methods for three different video sequences

	Seq. 1	Seq. 2	Seq. 3	Average PSNR	Rank
LMS	34.6	25.2	22.0	24,9	2
Bicubic	35.5	26.1	21.5	24,9	2
Edge Adaptive	34.5	24.7	20.5	23,7	4
Wavelet Interpolation	34.1	23.9	19.6	22,9	5
MRF Prediction	32.7	25.5	21.2	24,4	3
Exponential	35.5	26.2	21.5	24,9	2
Robust SR	35.2	28.5	23.5	26,9	1

Table 5.2. SSIM values of all methods for three different video sequences

	Seq. 1	Seq. 2	Seq. 3	Average SSIM	Rank
LMS	0.91	0.84	0.81	0.85	2
Bicubic	0.92	0.86	0.77	0.85	2
Edge Adaptive	0.90	0.82	0.72	0.81	3
Wavelet Interpolation	0.88	0.77	0.65	0.77	4
MRF Prediction	0.83	0.77	0.71	0.77	4
Exponential	0.91	0.86	0.77	0.85	2
Robust SR	0.89	0.87	0.82	0.86	1

The first cell in the first row of Table 5.3 gives the PSNR value of frame 1 for LMS method and was found to be 35.5 dB. The seventh cell in the first row of Table 5.3 is the average PSNR value and is found using the same methodology described in average video PSNR calculation. The eighth cell in each row of Table 5.3 gives the rank of the corresponding method. Average PSNR value in the seventh cell is used to determine this rank. According to average PSNR values, robust SR is found to be the most successful method.

Similarly, the first cell in the first row of Table 5.4 gives the SSIM value of frame 1 for LMS method and was found to be 0.92. The seventh cell in the first row of Table 5.4 is the average of the SSIM values of the six frames in a row-wise manner. The eighth cell in each row of Table 5.4 gives the rank of the corresponding method. Average value in the seventh cell is used to determine rank. According to average SSIM values, LMS method and robust SR are the most successful methods.

Error images for the 100^{th} frames of video sequences 1, 2 and 3 are given in Figure 5.5, Figure 5.6 and Figure 5.7, respectively. The error images were found by subtracting the processed image from the original image, taking the absolute value of the result and scaling by 10. Notice that the error images of robust SR method have the weakest magnitude in all three cases. This result is consistent with our PSNR and SSIM measurements.

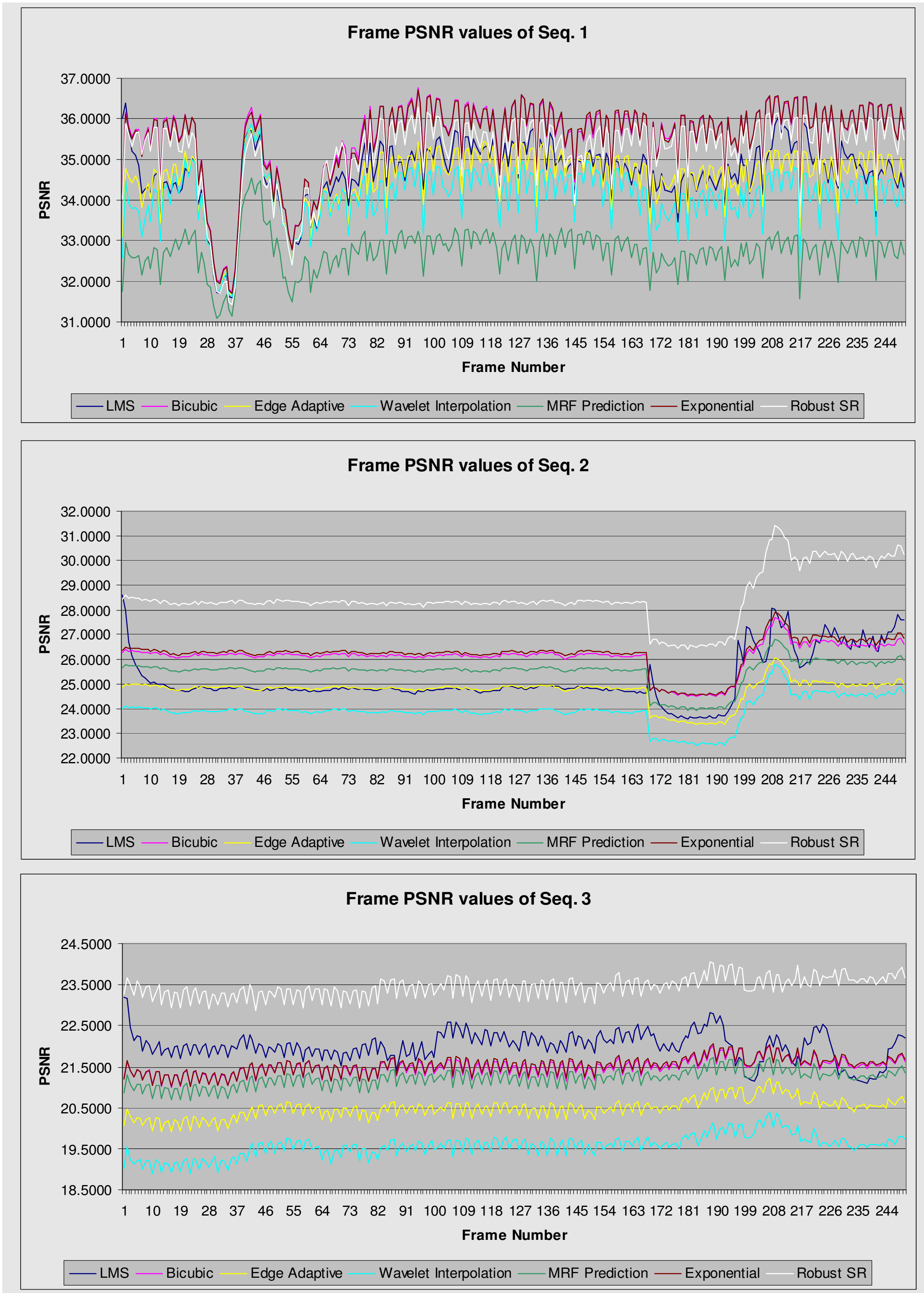


Figure 5.2. PSNR values of each frame in sequences 1, 2 and 3 for all methods

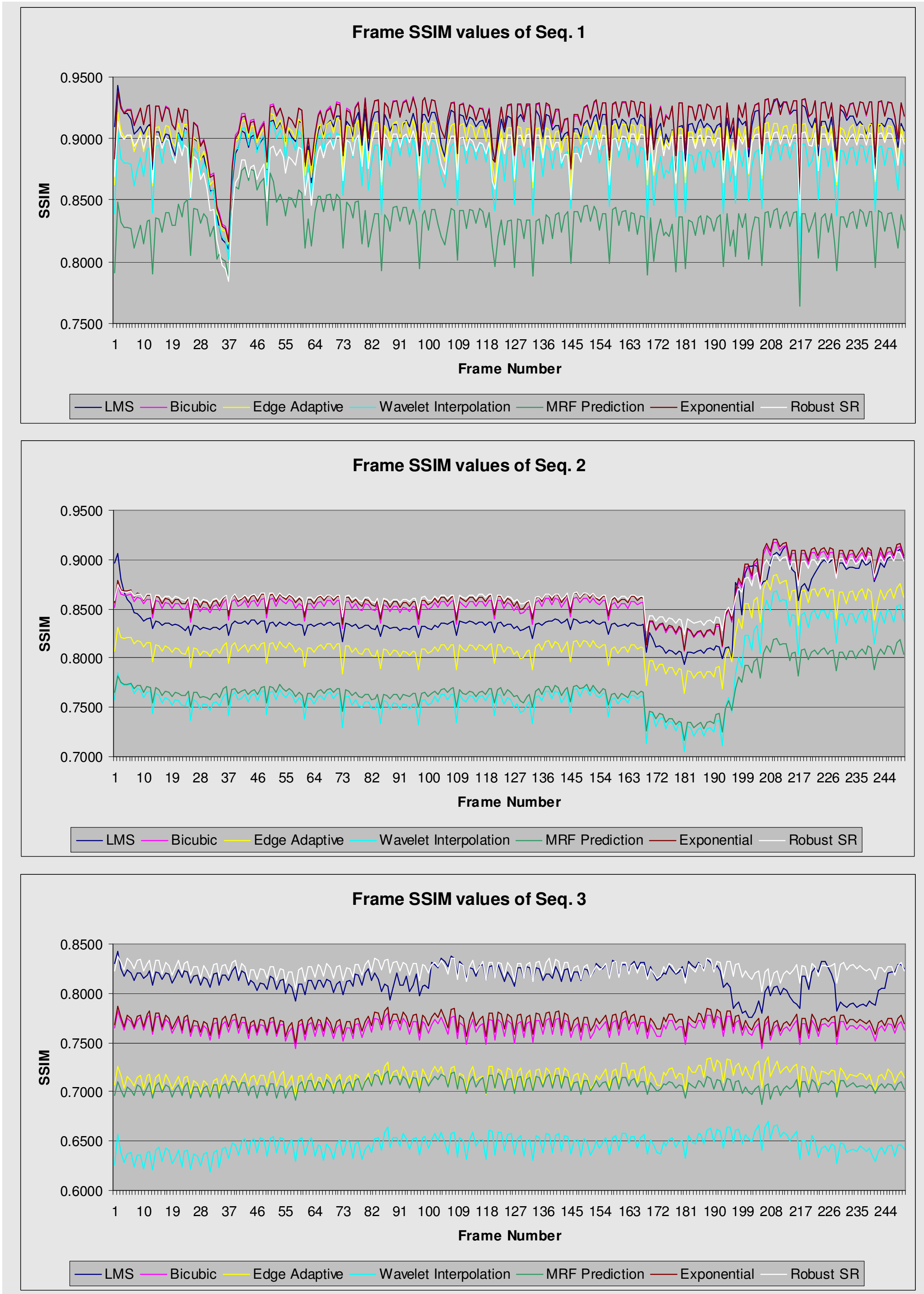


Figure 5.3. SSIM values of each frame in sequences 1, 2 and 3 for all methods

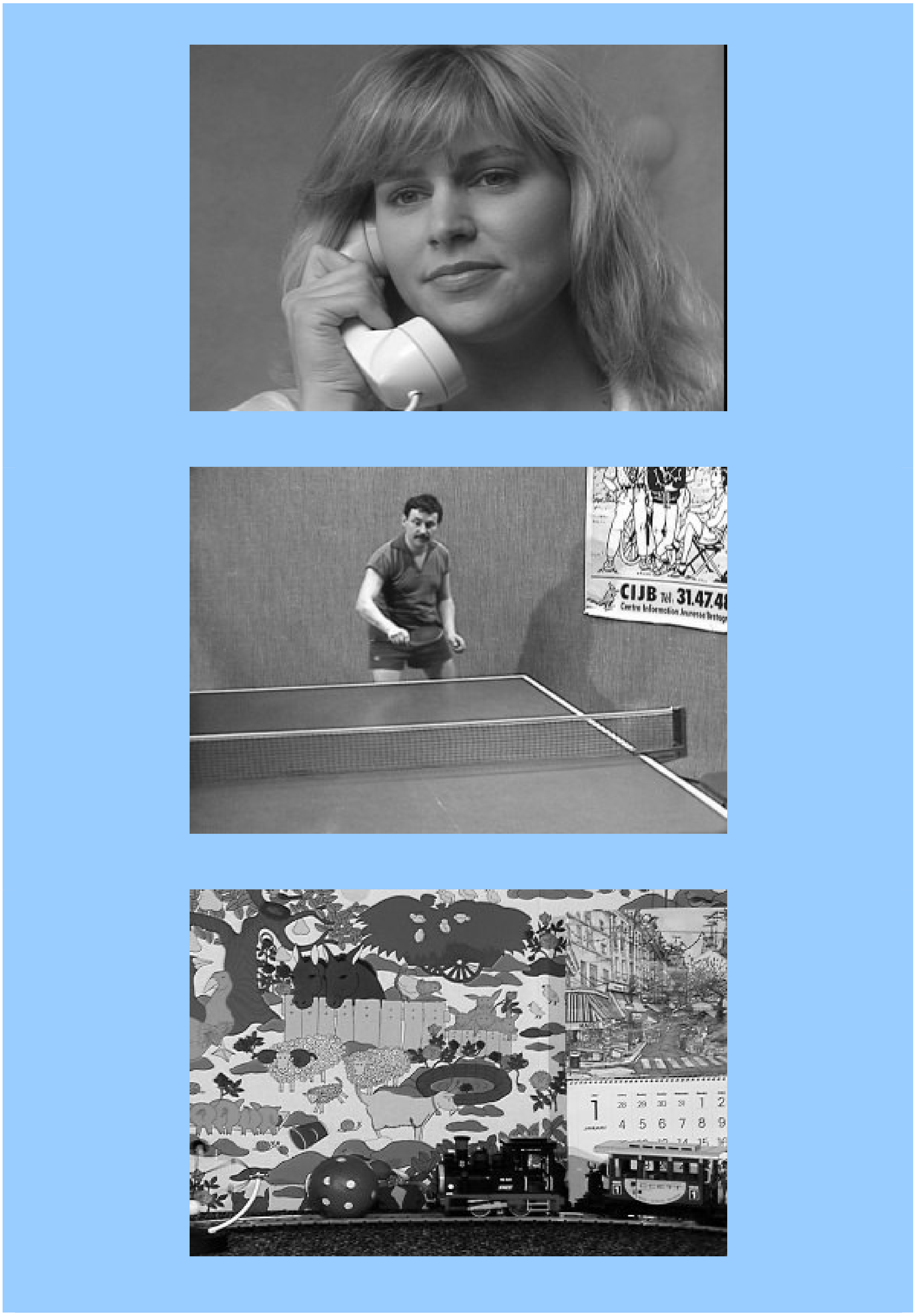


Figure 5.4. 100th frames of the 1st, 2nd and 3rd video sequences, respectively

Table 5.3. PSNR values of all methods for six different frame stills

	100th Frame of Seq. 1	200th Frame of Seq. 1	100th Frame of Seq. 2	200th Frame of Seq. 2	100th Frame of Seq. 3	200th Frame of Seq. 3	Average PSNR	Rank
LMS	35.5	34.1	24.8	27.2	22.3	21.2	24,9	4
Bicubic	36.5	35.2	26.1	26.4	21.5	21.5	24,9	4
Edge Adaptive	35.3	34.2	24.8	25.0	20.4	20.6	23,8	5
Wavelet Interpolation	34.8	33.6	23.9	24.3	19.5	19.8	23,0	6
MRF Prediction	33.1	32.4	25.6	25.7	21.2	21.2	25,7	3
Exponential	36.5	35.2	26.3	26.6	21.5	24.5	25,8	2
Robust SR	36.1	35.0	28.3	29.2	23.4	23.3	26,8	1

Table 5.4. SSIM values of all methods for six different frame stills

	100th Frame of Seq. 1	200th Frame of Seq. 1	100th Frame of Seq. 2	200th Frame of Seq. 2	100th Frame of Seq. 3	200th Frame of Seq. 3	Average SSIM	Rank
LMS	0.92	0.88	0.83	0.89	0.83	0.78	0.86	1.
Bicubic	0.93	0.89	0.85	0.89	0.77	0.77	0.85	2.
Edge Adaptive	0.92	0.87	0.81	0.86	0.73	0.73	0.82	3.
Wavelet Interpolation	0.9	0.85	0.76	0.82	0.66	0.67	0.78	4.
MRF Prediction	0.84	0.8	0.77	0.79	0.72	0.71	0.77	5.
Exponential	0.93	0.89	0.86	0.89	0.78	0.78	0.85	2.
Robust SR	0.91	0.87	0.86	0.88	0.83	0.83	0.86	1.

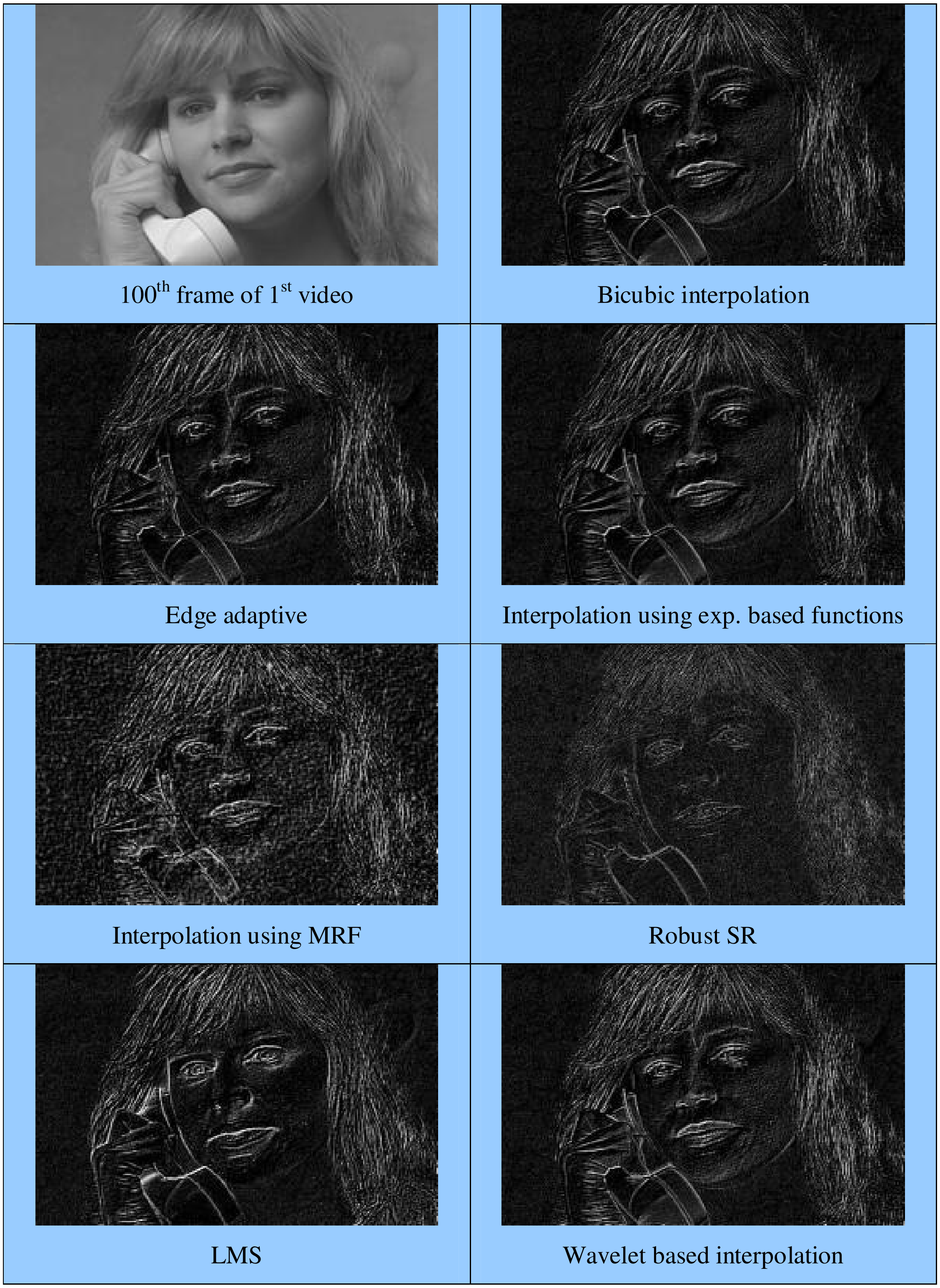


Figure 5.5. Error images of all methods for 100th frame of 1st video

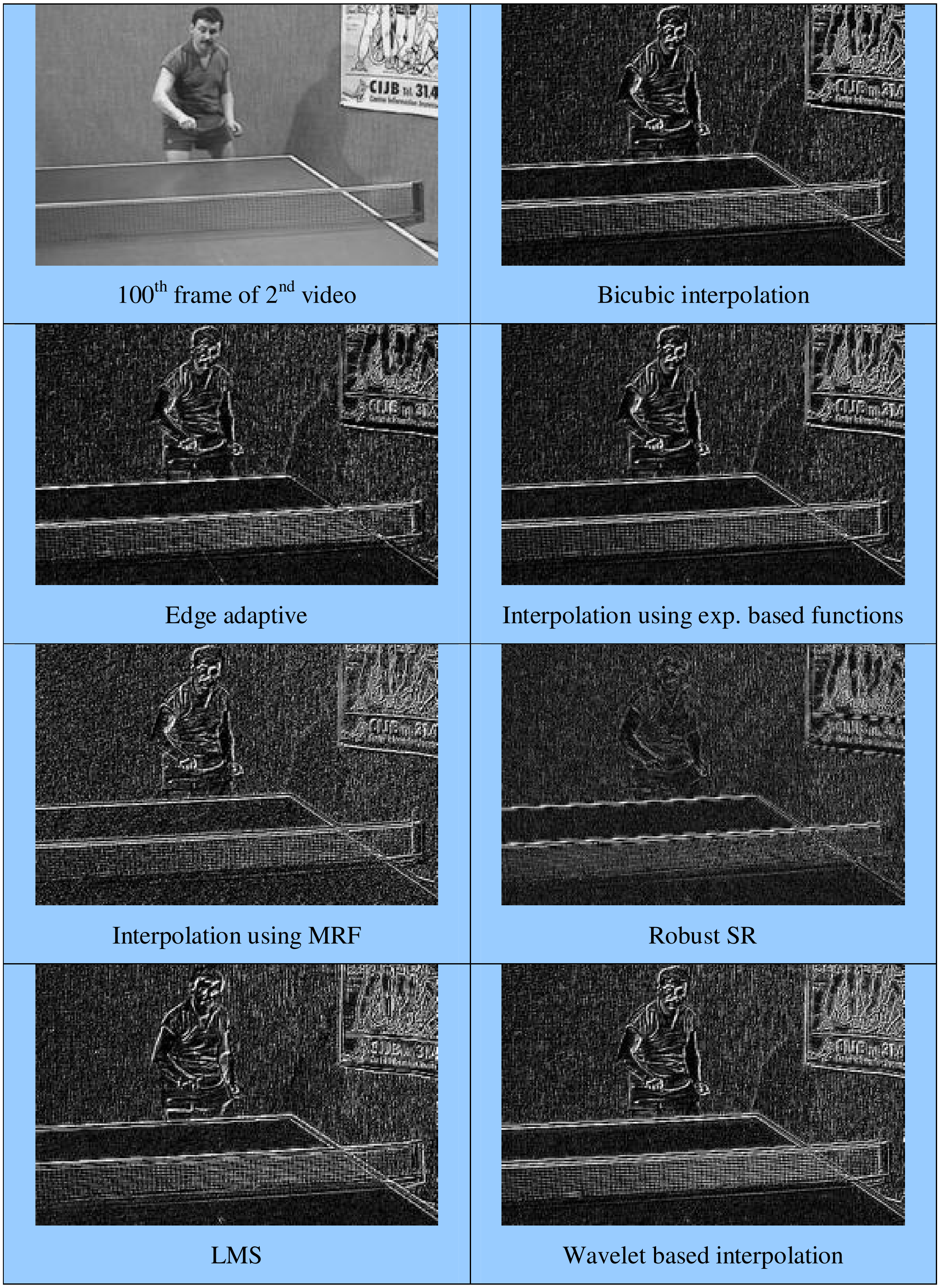


Figure 5.6. Error images of all methods for 100th frame of 2nd video

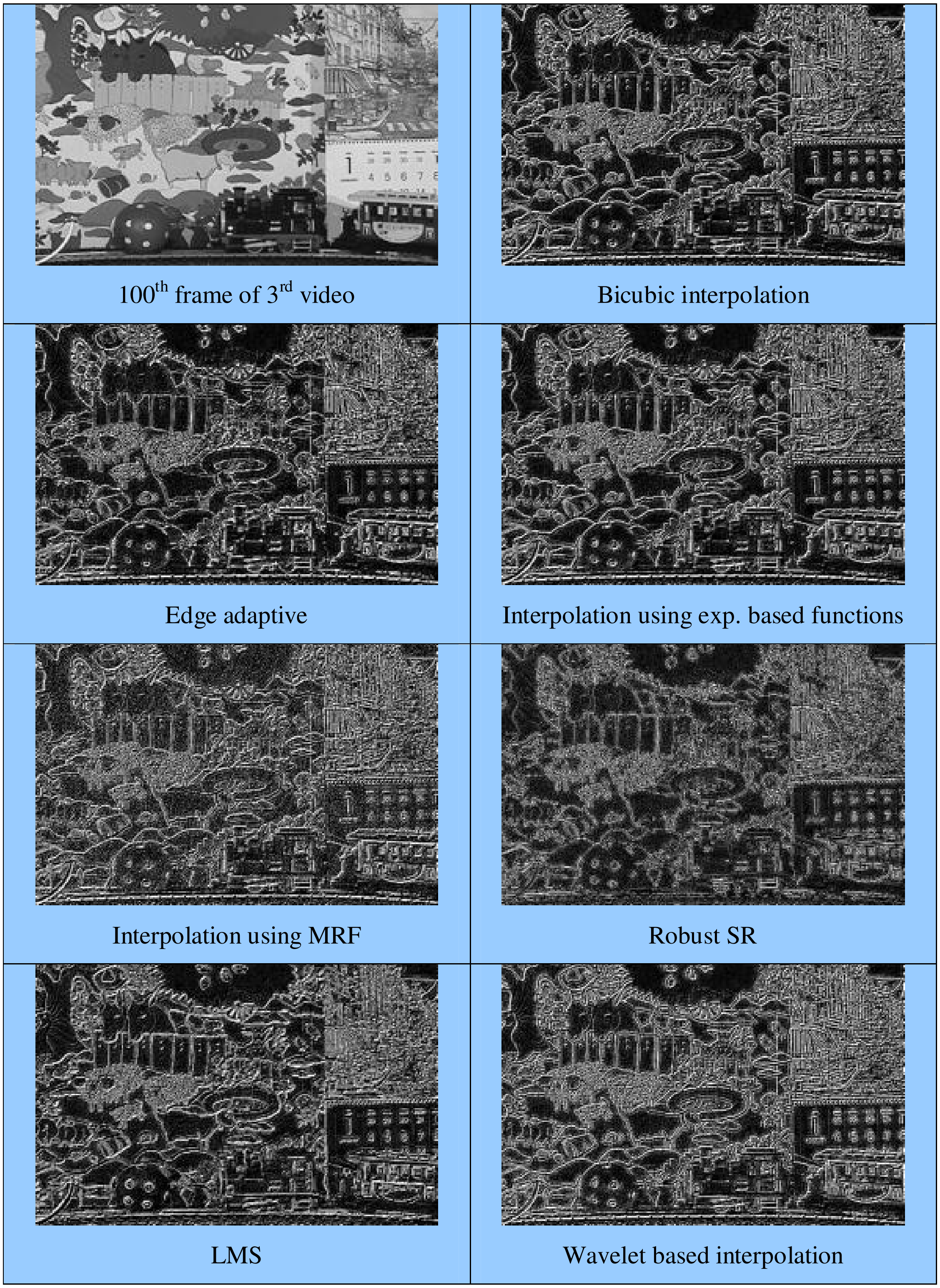


Figure 5.7. Error images of all methods for 100th frame of 3rd video

5.2. Experimental Setup

Subjectively perceived quality of the video sequences and frame stills produced by five different methods (LMS method, bicubic interpolation, wavelet based interpolation, edge adaptive interpolation, interpolation using wide sense Markov random fields) were tested as part of a psychovisual study using human observers. A total of 28 people took part in the experiment. Subjects had normal or corrected-to-normal vision. The ages of the subjects were in between 22 and 30. A typical experimental session on one subject lasted approximately forty five minutes. The experiments were performed in a dim room using a 1280x720 pixel resolution LCD display with 60 Hz refresh rate from a viewing distance of 30 cm.

The results were evaluated using “Two Alternative Forced Choice (2AFC)” paradigm which is illustrated in Figure 5.8. In 2AFC, the subject is presented with two alternatives and is asked to choose the better one in subjective video quality. The decision at the end of each comparison is 1 or 0. In a “method A vs. method B” comparison, if A wins, then the decision is 1, otherwise 0. There was no time pressure on the subjects and they could toggle back and forth between test pairs as much as they wanted.

In subjective tests, the decimated video sequences are resized to their original resolution (704x480) by five different methods. The subjects were prosecuted with six versions of each sequence, the original and the processed ones with five algorithms. The same method was also applied to frame stills. Thus the subjects had to evaluate comparatively 10 pairs (the number of 2-combination of 5 is 10) for each sequence and for each frame. The comparisons of 3 sequences result in 30 2AFC marks and the comparisons of 6 frames result in 60 2AFC marks. The results of these 90 comparisons are written to Excel sheets. The Excel sheet used in the experiment is shown in Figure 5.8. Notice that there are 10 decision boxes. One excel sheet is used for each video sequence and each frame still.

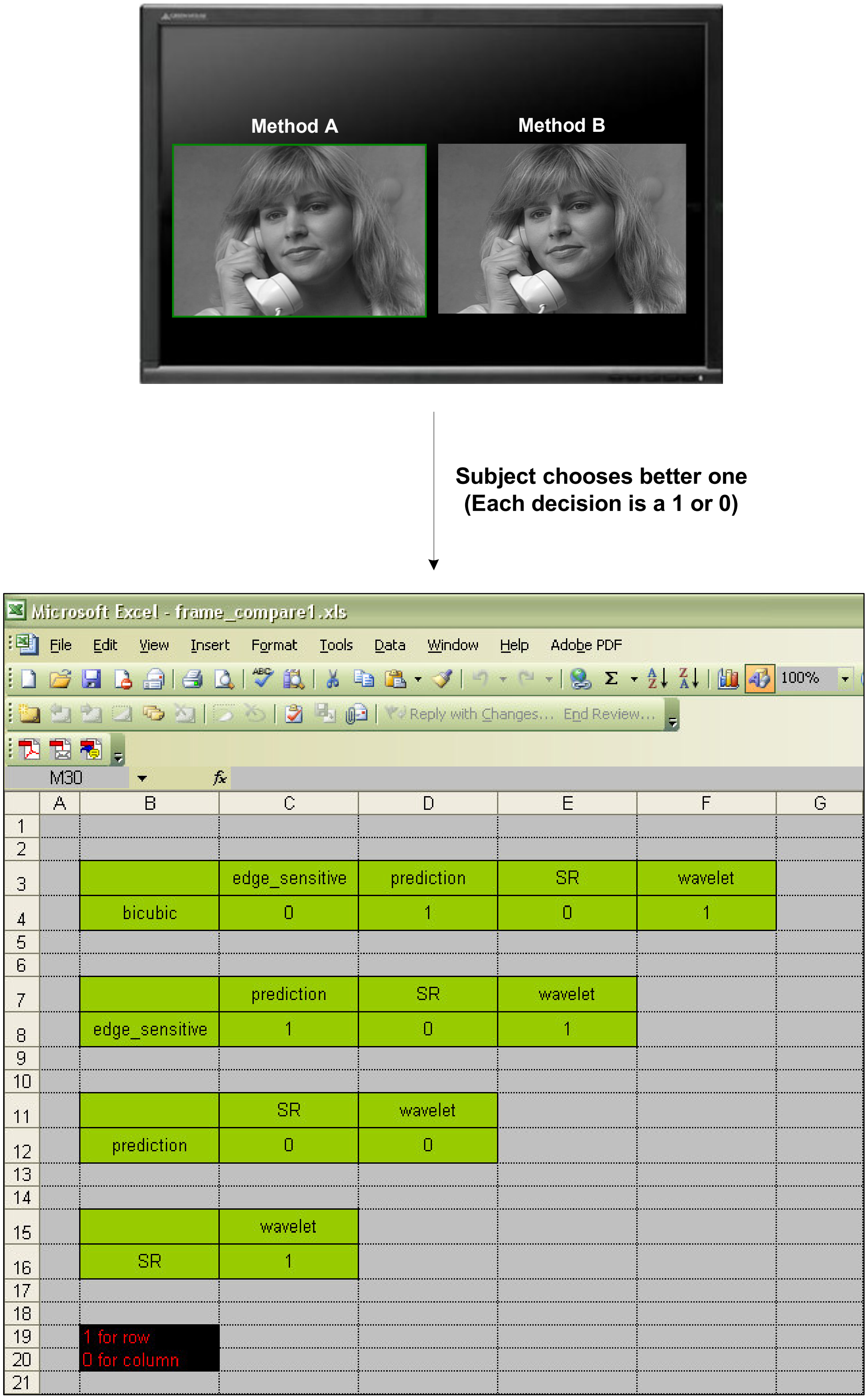


Figure 5.8. 2AFC test and the decisions on Excel sheet

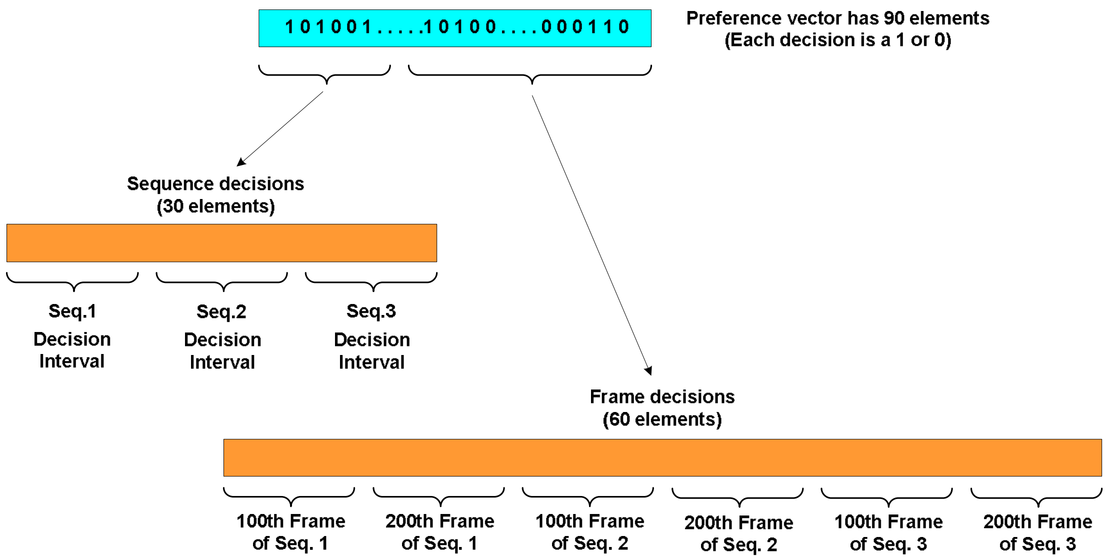


Figure 5.9. Structure of the preference vector

5.3. Results

5.3.1. Outlier Subjects

As the first step, preference vectors are created for each subject by concatenating his/her decisions for all 90 comparisons (i.e. 2AFC binary vectors). The structure of the preference vector is shown in Figure 5.9. The preference vector is composed of 30 video sequence votes and 60 frame still votes. There are decision intervals for each sequence and or each frame. These intervals include 10 bits and each bit is the decision of a “method A vs. method B” comparison. There were 28 preference vectors since the subject population was 28. Then, the mean of the preference vectors was found and was rounded to 0 or 1. After this, Hamming distance between each preference vector and the mean vector could be computed. The histogram of Hamming distances for 28 preference vectors is given in Figure 5.10.

We wanted to screen the data for outlier preference vectors using Hamming distance. Block diagram of outlier detection stage is given in Figure 5.11. If the Hamming distance of a preference vector from the group mean is outside the interquartile range, this means the preference vector is an outlier. In interquartile range method, Hamming distances of all vectors are sorted in an ascending order. First quartile Q_1, second quartile Q_2 (median) and third quartile Q_3 are found. Using quartile values, interquartile range (IQR) is computed using formula below

$$IQR = Q_3 - Q_1 \tag{5.1}$$

Finally, a preference vector is decreed as an outlier if its Hamming distance is outside the range

$$[Q_1 - k.IQR, Q_3 + k.IQR] \tag{5.2}$$

where $k=1.5$ in our experiments. The result of data screening revealed that only 1 participant (with Hamming distance of 29) was qualified as an outlier.

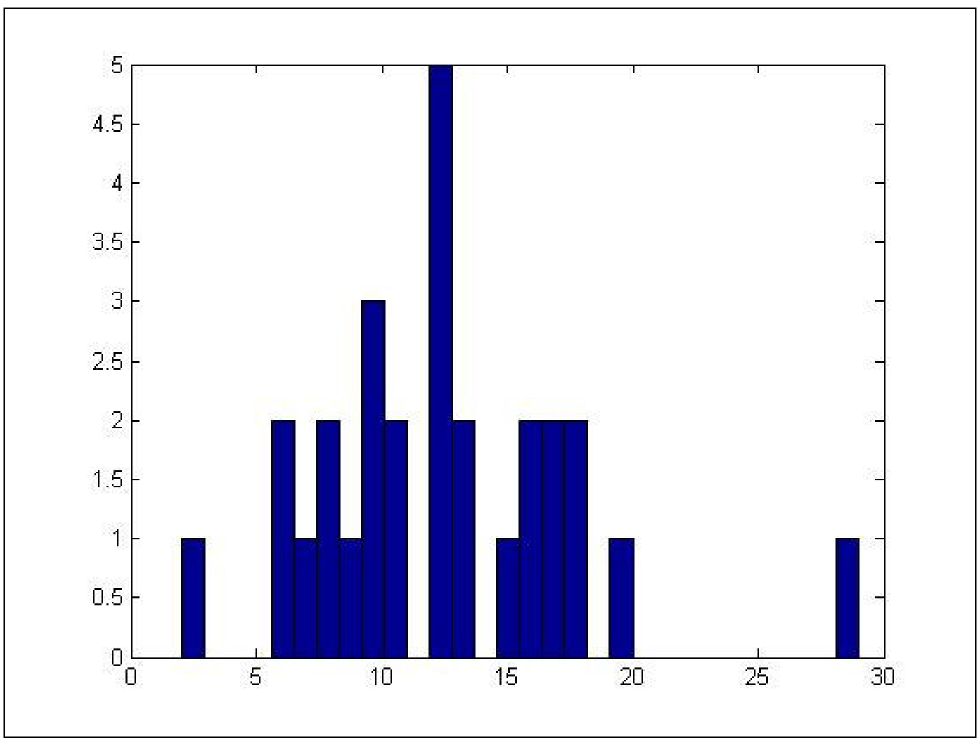


Figure 5.10. Histogram of Hamming distances for 28 preference vectors

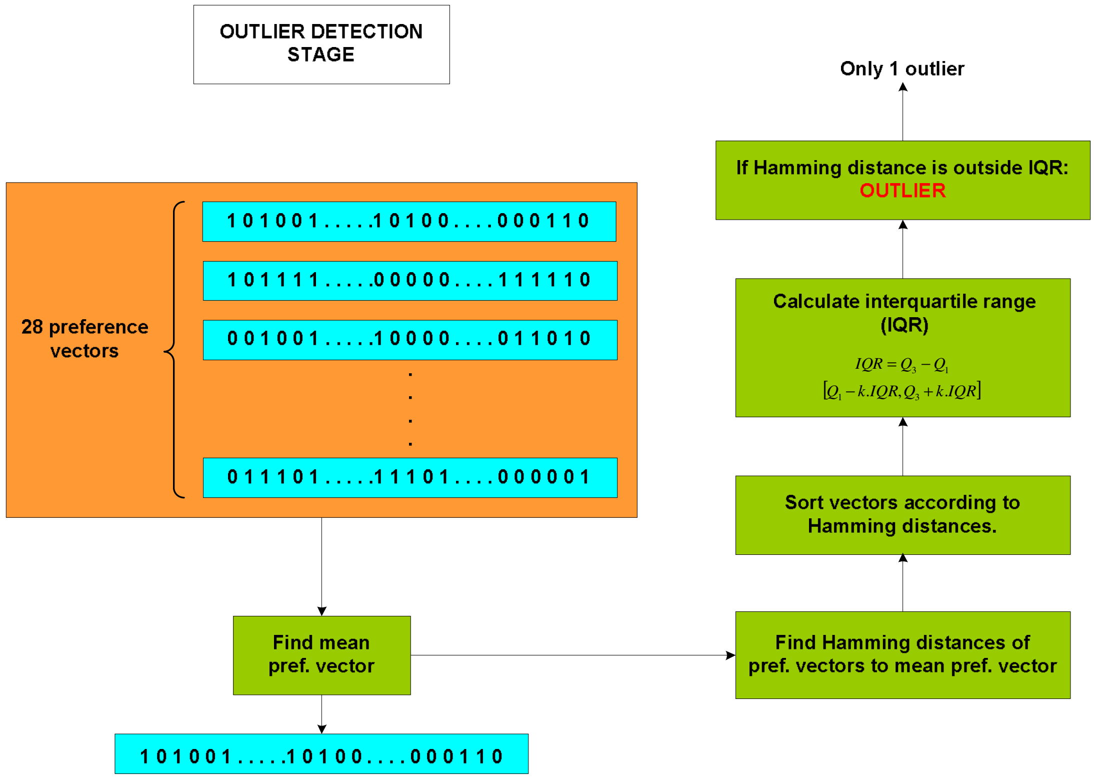


Figure 5.11. Block diagram of outlier detection stage

5.3.2. Computation of Subjective Scores

After getting rid of the outlier preference vector, there remain 27 preference vectors. These vectors contain binary digits and each binary digit is the decision of a “method A vs. method B” comparison. For the sake of clarity, we had better convert these bits into decimal scores. Figure 5.12 shows the computation of preference scores for video sequences and frame stills.

Subjective video sequence scores and subjective frame still scores were computed separately. Recall that each preference vector can be divided into two parts as sequence decisions and frame decisions. Sequence decisions were used to compute subjective sequence scores and frame decisions were used to compute frame scores. In a “method A vs. method B” comparison, we count how many times A wins in sequence decisions (or frame decisions) to find the subjective sequence score (or subjective frame score) of A for that comparison. Then the score is normalized between 0 and 100.

The preference scores for video sequences and frame stills are shown in Table 5.5 and Table 5.6, respectively. For example, the second cell in the first row of Table 5.5 indicates that LMS method is preferred over the bicubic 75.3% percent of the time, while the third cell indicates that it is overwhelmingly preferred over the edge-adaptive method, and so on. If we sum and average these scores row-wise, we get the total and average preference scores of a method over all its competitors. The rank orders of the methods for video sequences and frame stills are given in the rightmost columns of Table 5.5 and Table 5.6, respectively.

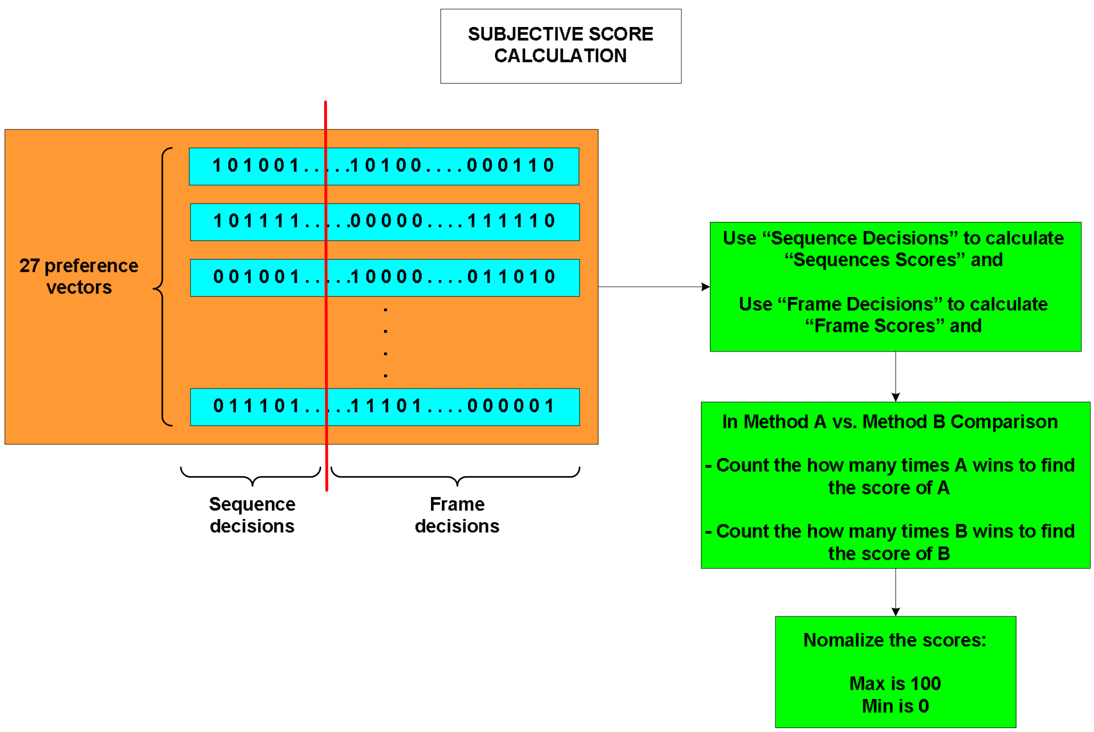


Figure 5.12. Computation of preference scores for video and still frames

Table 5.5. Rank and subjective score of each method for video sequences

	LMS	Bicubic	Edge Adapt.	Wavelet Interp.	MRF Pre.	Total Score	Average Score	Rank
LMS		75.3	96.3	87.7	98.8	358	89.5	1
Bicubic	24.7		92.6	67.9	92.6	277.8	69.4	2
Edge Adapt.	3.7	7.4		11.1	58.0	80.2	20.1	4
Wavelet Interp.	12.3	32.1	88.9		91.4	224.7	56.2	3
MRF Pre.	1.2	7.4	42.0	8.6		59.3	14.8	5

Table 5.6. Rank and subjective score of each method for frame stills

	LMS	Bicubic	Edge Adapt.	Wavelet Interp.	MRF Pre.	Total Score	Average Score	Rank
LMS		97.5	96.3	90.1	98.8	382.7	95.7	1
Bicubic	2.5		85.2	46.3	88.9	222.8	55.7	3
Edge Adapt.	3.7	14.8		19.1	56.2	93.8	23.5	4
Wavelet Interp.	9.9	53.7	80.9		85.8	230.2	57.6	2
MRF Pre.	1.2	11.1	43.8	14.2		70.4	17.6	5

5.3.3. Comparison of Subjective Video and Subjective Frame Still Scores

First, we address the question as to whether one can asses HR quality alternatively with frame stills and video sequences and whether their scores are in agreement. Conversely, are there any objectionable defects that become evident only in video but that is not observable in frame still? Spearman's rank correlation coefficients were computed between the subjective decisions in frame still and video cases.

Spearman's rank correlation coefficient is a measure of correlation between two data vectors. Let each vector have n elements. The elements of two vectors are sorted in an ascending order by the first vector. The rank of each element is determined. This results in two rank vectors. MSE between these two rank vectors is computed. Finally, Spearman's rank correlation coefficient between two vectors is determined using the formula below

$$\rho = 1 - \frac{6MSE}{n\left(n^2 - 1\right)} \tag{5.3}$$

Figure 5.13 shows the block diagram illustrating the computation of Spearman's rank correlation coefficients between subjective sequence scores and subjective frame still scores. Recall that in the decision interval of each sequence and each frame, there is one bit related to "method A vs. method B" comparison. Combining these bits columnwise as shown in Figure 5.13, we get 3 vectors in sequence case and 6 vectors in frame case. Then, 3 sequence vectors are averaged and the resultant vector represents the decisions related to "method A vs. method B" comparison in sequence case. Also, 6 sequence vectors are averaged and the resultant vector represents the decisions related to "method A vs. method B" comparison in still frame case. The average sequence and the average still frame vectors are used to compute Spearman's rank correlation coefficient between subjective sequence scores and subjective still frame scores for "method A vs. method B" comparison. The coefficients for each comparison are given in Table 5.7. If correlation coefficient is 1, then the decisions for that comparison are perfectly correlated in frame still and video cases. According to Table 5.7, we can say that video and frame decisions are highly correlated.

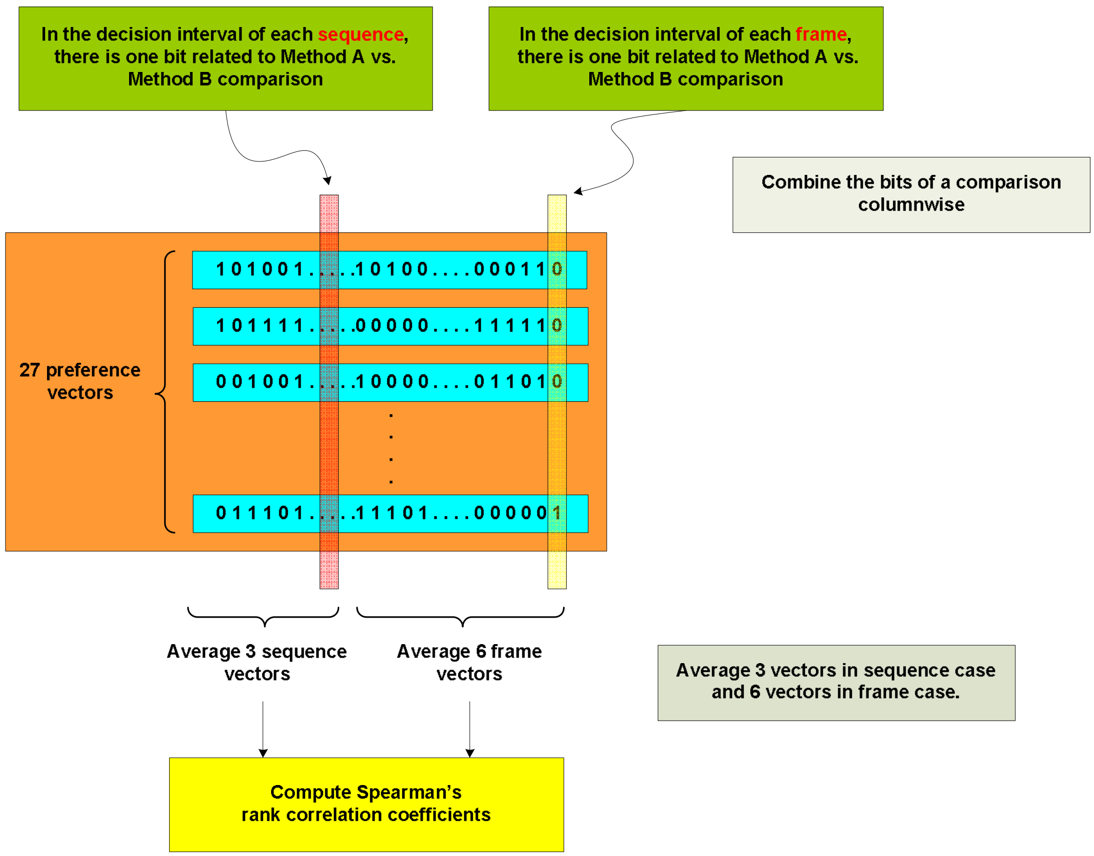


Figure 5.13. Computation of Spearman's rank correlation coefficients between subjective video and subjective frame stills

Table 5.7. Spearman's rank correlation coefficients between subjective video scores and subjective frame scores for each comparison

	LMS	Bicubic	Edge Adap.	Wavelet	MRF Pre.
LMS		1.0	1.0	1.0	1.0
Bicubic	1.0		0.9994	0.5452	1.0
Edge Adap.	1.0	0.9994		0.6624	0.4835
Wavelet	1.0	0.5452	0.6624		0.6197
MRF Pre.	1.0	1.0	0.4835	0.6197	

The least coefficient belongs to “edge adaptive – MRF prediction” comparison. In this comparison, the decision changes between frame still and video cases.

5.3.4. Comparison of Subjective and Objective Scores

In this section, we intend to investigate the degree of agreement between subjective preferences and objective measurements. Spearman’s rank correlation coefficients for “subjective scores – PSNR measurements” and “subjective scores – SSIM measurements” were computed. Details of the procedure are illustrated in Figure 5.14. As in Section 5.3.1, the preference vectors were averaged and the elements of the average vector were rounded to 0 or 1. In order to reduce the objective measure comparisons to a binary vector, we used the following strategy: if A method has a higher PSNR value than B method (for a frame still or video), then the decision is 1 otherwise 0. SSIM vector is created using the same methodology. At this point, there is one 90 element binary vector for PSNR measurements, SSIM measurements and subjective scores. These three vectors were used while computing Spearman’s rank correlation coefficients between subjective scores and objective measurements.

Using all decisions for all methods, the Spearman’s rank correlation coefficient in “subjective scores – PSNR measurements” case is 0.8676 and the coefficient in “subjective scores – SSIM measurements” case is 0.8875. This means, using all decisions, the correlation in “subjective scores – SSIM measurements” case and the correlation in “subjective scores – PSNR measurements” case are nearly the same.

While computing correlation coefficient for a single comparison, the related parts of the three vectors (subvectors) were extracted and used. Subvector extraction process is shown in Figure 5.15. There is one bit related to “method A vs. method B” comparison in the decision interval of each video and each frame. Combining these bits in each preference vector in a rowwise manner, we get the related subvector for that comparison. One subvector is extracted for each comparison from PSNR, SSIM and mean preference vectors. Then these three subvectors are used to determine the correlation coefficients for that comparison. Method to method comparisons are given in Table 5.8 and Table 5.9.

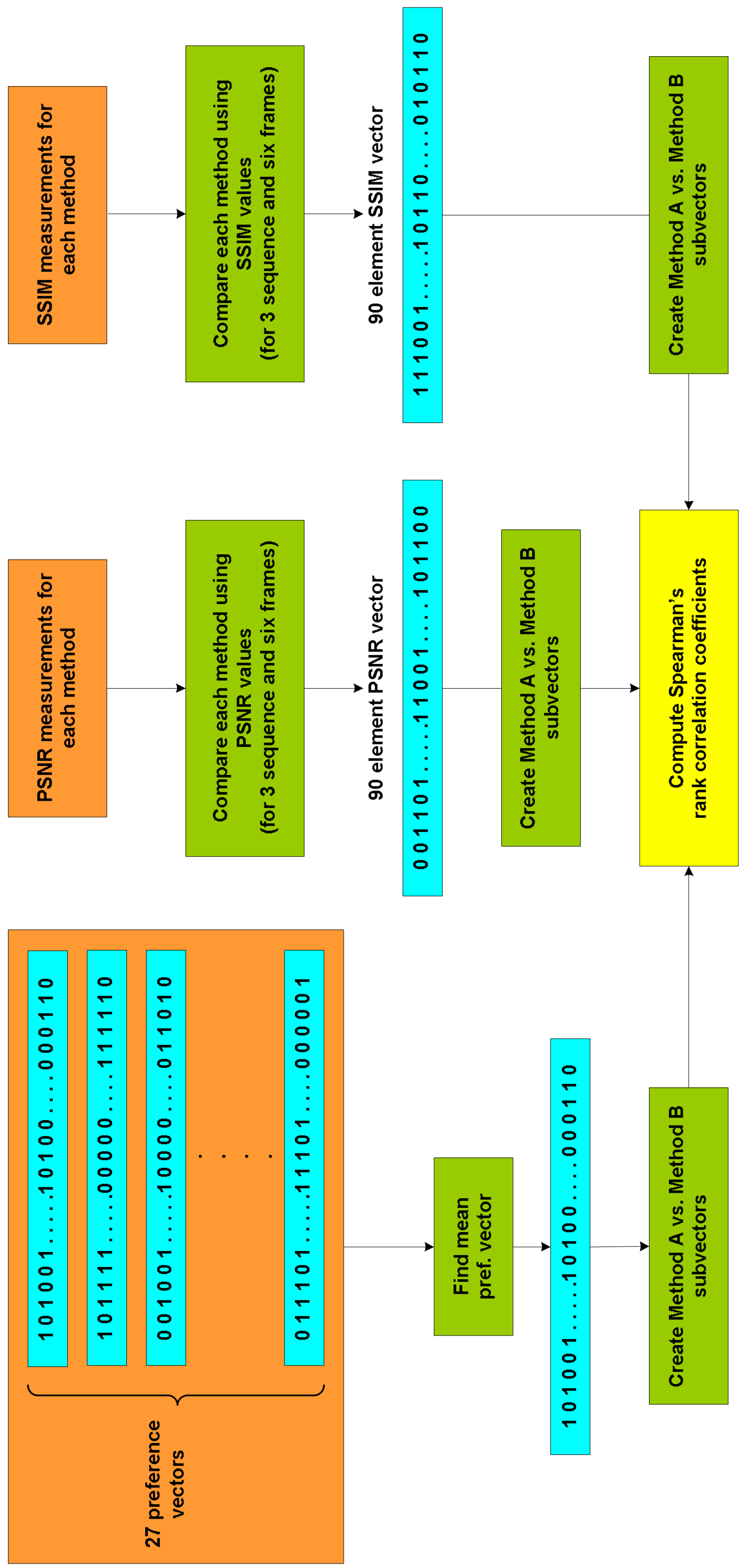


Figure 5.14. Computation of Spearman's rank correlation coefficients between subjective scores and objective measurements

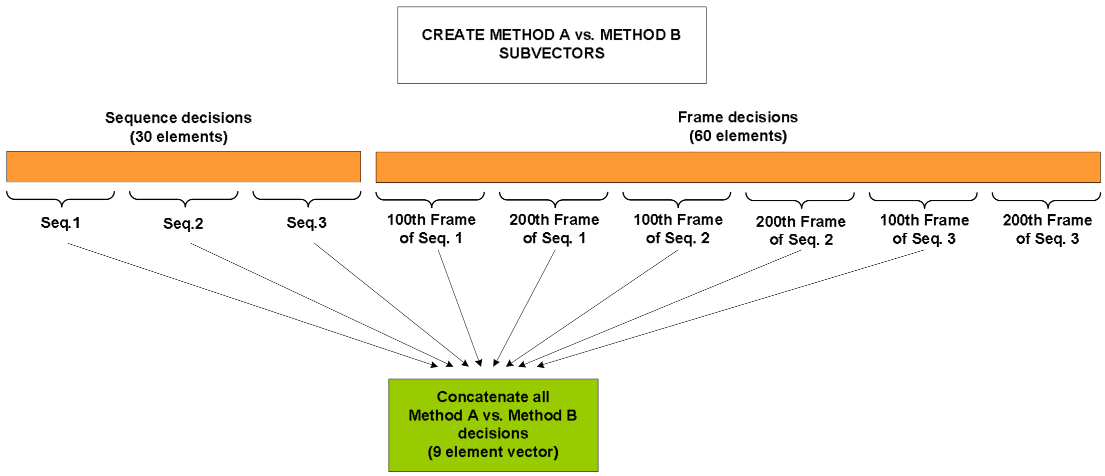


Figure 5.15. Extraction of subvectors for each "method A vs. method B" comparison

As can be understood from method to method comparisons, the correlation in "subjective scores – SSIM measurements" is almost always is equal to 1 which means perfect correlation between subjective scores and SSIM measurements. Also, the correlation in "subjective scores – SSIM measurements" is always equal to or higher than the correlation in "subjective scores – PSNR measurements". We can conclude that SSIM measurements are highly correlated with the decisions of human visual system and SSIM models the human visual system better than PSNR do.

One exceptional case is "LMS – Bicubic" comparison. The least correlation coefficient in both tables belong to "LMS – Bicubic" comparison. The result of bicubic method resembles the original input. So, bicubic method gets high objective measurements (PSNR and SSIM results). On the other hand, the result of the LMS method is better than the original inputs, because of the fact that LMS is a multi-frame method and LR samples coming from neighboring frames increases the quality of the SR estimate. For this reason, LMS gets high subjective scores.

Table 5.8. Spearman's rank correlation coefficients between subjective scores and PSNR measurements

	LMS	Bicubic	Edge Adap.	Wavelet	MRF Pre.
LMS		**0.3**	**0.75**	**1.0**	**0.6667**
Bicubic	**0.3**		**1.0**	**1.0**	**1.0**
Edge Adap.	**0.75**	**1.0**		**1.0**	**0.9**
Wavelet	**1.0**	**1.0**	**1.0**		**0.8667**
MRF Pre.	**0.6667**	**1.0**	**0.9**	**0.8667**	

Table 5.9. Spearman's rank correlation coefficients between subjective scores and SSIM measurements

	LMS	Bicubic	Edge Adap.	Wavelet	MRF Pre.
LMS		**-0.15**	**1.0**	**1.0**	**1.0**
Bicubic	**-0.15**		**1.0**	**1.0**	**1.0**
Edge Adap.	**1.0**	**1.0**		**1.0**	**1.0**
Wavelet	**1.0**	**1.0**	**1.0**		**0.8667**
MRF Pre.	**1.0**	**1.0**	**1.0**	**0.8667**	

6. CONCLUSION

In this work, a literature survey about different SR reconstruction methods is given in section 2. Spatial domain and frequency domain SR reconstruction methods are covered in detail. Some well known analytical interpolation methods are given in section 3. Two SR algorithms (robust SR and LMS method) were compared with 5 analytical interpolation methods (bicubic interpolation, wavelet based interpolation, edge adaptive interpolation, interpolation using wide sense Markov random fields and interpolation using exponential based kernels). The results of these seven methods were evaluated using the metrics described in section 4. According to PSNR values, robust SR is the most successful method for sequences (Table 5.1) and for frames (Table 5.3). On the other hand, SSIM calculations revealed that robust SR is the most successful method for sequences (Table 5.2), robust SR and LMS methods are the most successful methods for frames (Table 5.4).

Also, the results of five methods (LMS method, bicubic interpolation, wavelet based interpolation, edge adaptive interpolation, interpolation using wide sense Markov random fields) were compared subjectively using a group of subjects. According to subjective scores, LMS method is better than all the interpolation methods for both sequence and frame cases (refer to Table 5.7 and Table 5.8). The correlation between the subjective and objective evaluation results was inspected in section 5.3.3. We can say that subjective decisions are highly correlated with objective measurements. Using all decisions for all methods, the Spearman's rank correlation coefficient between subjective scores and SSIM measurements are nearly equal to the correlation between subjective scores and PSNR measurements. But in method vs. method comparisons, SSIM models human visual system better than PSNR do.

REFERENCES

1. Freeman, W. T., T. R. Jones and E. Pasztor, “Example-based super-resolution”, *IEEE Computer Graphics and Applications*, pp. 56–65, Mar. 2002.

2. Bishop, C. M., A. Blake and B. Marthi, “Super-resolution enhancement of video”, *in Proceedings of the IEEE Artificial Intelligence and Statistics*, 2003.

3. Park, S. C., M. K. Park and M. G. Kang, “Super-Resolution Image Reconstruction: A Technical Overview”, *IEEE Signal Processing Magazine*, Vol. 20, pp. 21-36, May 2003.

4. Aizawa, K., T. Komatsu and T. Saito, “Very High Resolution Image Acquisition Through Image Reconstruction From Lower Resolution Images Taken With Multiple Cameras”, *Singapore ICCS/ISITA '92. 'Communications on the Move'*, Vol. 2, pp. 639–643, Nov. 1992.

5. Aizawa, K., T. Komatsu and T. Saito, “A scheme for acquiring very high resolution images using multiple cameras”, *in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, San Francisco,* vol. 3, pp. 289–292, Mar. 1992.

6. Komatsu, T., K. Aizawa, T. Igarashi and T. Saito, “Signal-processing based method for acquiring very high resolution images with multiple cameras and its theoretical analysis”, *IEEE Proceedings I*, Vol. 140, pp. 19–25, Feb. 1993.

7. Landweber, L., “An iteration formula for Fredholm integral equations of the first kind”, *Amer. J. Math.*, Vol. 73, pp. 615-624, 1951.

8. Saito, T., T. Komatsu and K. Aizawa, “An image processing algorithm for a super high definition imaging scheme with multiple different-aperture cameras”, *in*

Proceedings of the IEEE International Conference of Acoustics, Speech and Signal Processing, Adelaide, Australia, Vol. 5, pp. 45–48, Apr. 1994.

9. Nakazawa, Y., T. Komatsu and T. Saito, "High resolution image acquisition based on temporal integration with hierarchical estimation of image warping", *in Proceedings of IEEE International Conference on Image Processing, Washington, DC,* Vol. 3, pp. 244–247, 1995.

10. Nakazawa, Y., T. Saito, T. Sekimori and K. Aizawa, "Two approaches for image-processing based high resolution image acquisition", *in Proceedings of the IEEE International Conference on Image Processing, Austin, TX*, Vol. 3, pp. 147–151, 1994.

11. Keren, D., S. Peleg and R. Brada, "Image sequence enhancement using sub-pixel displacements", *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 742-746, June 1988.

12. Shah, N. R. and A. Zakhor, "Multiframe spatial resolution enhancement of color video", *in Proceedings of the IEEE International Conference on Image Processing, Lausanne, Switzerland*, Vol. 1, pp. 985–988, Sept. 1996.

13. Shah, N. R. and A. Zakhor, "Resolution Enhancement of Color Video Sequences", *IEEE Transactions on Image Processing*, Vol. 8, Issue 6, pp. 879 – 885, June 1999.

14. Irani, M. and S. Peleg, "Super Resolution From Image Sequences", *Proceedings of the10th International Conference on Pattern Recognition*, Vol. 2, pp. 115–120, June 1990.

15. Schultz, R. R. and R. L. Stevenson, "A Bayesian approach to image expansion for improved definition", *IEEE Transactions on Image Processing*, Vol. 3, pp. 233–242, 1994.

16. Schultz, R. R. and R. L. Stevenson, “Improved definition image expansion”, *in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, San Francisco*, Vol. 3, pp. 173–176, Mar. 1992.

17. Schultz, R. R. and R. L. Stevenson, “Extraction of High-Resolution Frames from Video Sequences”, *IEEE Transactions on Image Processing*, Vol. 5, pp. 996–1011, 1996.

18. Schultz, R. R. and R. L. Stevenson, “Improved Definition Video Frame Enhancement”, *in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Detroit, MI*, Vol. 4, pp. 2169–2172, May 1995.

19. Schultz, R. R. and R. L. Stevenson, “Bayesian Estimation of Subpixel-Resolution Motion Fields and High-Resolution Video Stills”, *in Proceedings of the IEEE International Conference on Image Processing, Santa Barbara, CA,* Vol. 3, pp. 62–65, Oct. 1997.

20. Bouman, C. and K. Sauer, “A Generalized Gaussian Image Model for Edge-Preserving MAP Estimation”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 2, no. 3, pp. 296–310, July 1993.

21. Hardie, R. C., K. J. Barnard and E. E. Armstrong, “Joint MAP Registration and High-Resolution Image Estimation Using a Sequence of Undersampled Images”, *IEEE Transactions on Image Processing*, Vol. 6, no. 12, pp. 1621–1633, Dec. 1997.

22. Elad, M. and A. Feuer, “Restoration of a Single Superresolution Image from Several Blurred, Noisy, and Undersampled Measured Images”, *IEEE Transactions on Image Processing*, Vol. 6, no.12, pp. 1646-1658, Dec. 1997.

23. Elad, M. and A. Feuer, “Super-Resolution Reconstruction of an Image”, in *Proceedings of the 19th IEEE Conference, Jerusalem, Israel,* pp. 391-394, Nov. 1996.

24. Fard, P. O. and H. Stark, "Tomographic image reconstruction using the theory of convex projections", *IEEE Transactions on Medical Imaging*, Vol. 7, no. 1, pp. 45–58, Mar. 1988.

25. Tekalp, A. M., M. K. Ozkan and M. I. Sezan, "High-resolution image reconstruction from lower-resolution image sequences and space-varying image restoration", *in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, San Francisco, CA*, Vol. 3, pp. 169–172, 1992.

26. Patti, A. J., M. I. Sezan and A. M. Tekalp, "High-resolution image reconstruction from a low-resolution image sequence in the presence of time-varying motion blur", *in Proceedings of the IEEE International Conference on Image Processing, Austin, TX,* Vol. 1, pp. 343–347, 1994.

27. Patti, A. J., M. I. Sezan and A. M. Tekalp, "High resolution standards conversion of low resolution video", *in Proceedings of the IEEE International Conference of Acoustics, Speech and Signal Processing, Detroit, MI*, Vol. 4, pp. 2197–2200, 1995.

28. Patti, A. J., M. I. Sezan and A. M. Tekalp, "Superresolution Video Reconstruction with Arbitrary Sampling Lattices and Nonzero Aperture Time", *IEEE Transactions on Image Processing*, Vol. 6, no. 8, pp. 1064–1076, Aug. 1997.

29. Tom, B. C. and A. K. Katsaggelos, "Resolution enhancement of video sequences using motion compensation", *in Proceedings of the IEEE International Conference on Image Processing, Lausanne, Switzerland,* Vol. 1, pp. 713–716, Sept. 1996.

30. Hong, M. C., M. G. Kang and A. K. Katsaggelos, "An iterative weighted regularized algorithm for improving the resolution of video sequences", *in Proceedings of the IEEE International Conference on Image Processing, Santa Barbara, CA,* Vol. 2, pp. 474–477, Oct. 1997.

31. Patanavijit, V. and S. Jitapunkul, "A Lorentzian Stochastic Estimation for a Robust Iterative Multiframe Super-Resolution Reconstruction with Lorentzian-Tikhonov

Regularization", *EURASIP Journal on Advances in Signal Processing*, vol 2007, id 34821, pp. 609-616, 2007.

32. Haykin, S., *Adaptive Filter Theory*, Prentice-Hall Inc., 1-st Edition, 1986.

33. Elad, M. and A. Feuer, "Super-resolution restoration of continuous image sequence using the LMS algorithm", *in Eighteenth Convention of Electrical and Electronics Engineers in Israel,* pp. 2.2.5/1-2.2.5/5, 7-8 March 1995.

34. Elad, M. and A. Feuer, "Superresolution restoration of an image sequence: adaptive filtering approach", *in IEEE Transactions on Image Processing*, Vol. 8, Issue 3, pp. 387–395, March 1999.

35. Elad, M. and A. Feuer, "Super-resolution reconstruction of continuous image sequences", *in Image Processing, ICIP 1999, Proceedings on International Conference*, Vol. 3, pp. 459–463, 24-28 Oct. 1999.

36. Costa, G. H. and J. C. M. Bermudez, "Statistical Analysis of the LMS Algorithm Applied to Super-Resolution Image Reconstruction", *IEEE Transactions on Signal Processing*, Vol. 55, Issue 5, Part 2, pp. 2084–2095, May 2007.

37. Costa, G. H. and J. C. M. Bermudez, "Informed Choice of the LMS Parameters in Super-Resolution Video Reconstruction Applications", *IEEE Transactions on Signal Processing*, Vol. 56, Issue 2, pp. 555–564, Feb. 2008.

38. Kaltenbacher, E. and R. C. Hardie, "High-resolution infrared image reconstruction using multiple low resolution aliased frames", *in Proceedings of the IEEE National Aerospace Electronics Conference (NAECON), Dayton, OH*, Vol. 2, pp. 702–709, May 1996.

39. Kim, S. P. and W. Su, "Recursive high-resolution reconstruction of blurred multiframe images", *in Proceedings of the IEEE International Conference on*

Acoustics Speech and Signal Processing, Toronto, Canada, Vol. 4, pp. 2977–2980, May 1991.

40. Kim, S. P. and W. Y. Su, "Recursive high-resolution reconstruction of blurred multiframe images", *IEEE Transactions on Image Processing*, Vol. 2, pp. 534–539, Oct. 1993.

41. Keys, R., "Cubic convolution interpolation for digital image processing", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 29, pp. 1153–1160, Dec 1981.

42. Reichenbach, S. E. and F. Geng, "Two-dimensional cubic convolution", *IEEE Transactions on Image Processing*, Vol. 12, pp. 857–865, Aug. 2003.

43. Hou, H. S., H. C. Andrews, "Cubic splines for image interpolation and digital filtering", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 26, pp. 508–517, Dec 1978.

44. Turkowski, K., "Filters for Common Resampling Tasks**",** *Apple Computer*, Apr.1990

45. Chang, S. G., Z. Cvetkovic, M. Vetterli, "Resolution enhancement of images using wavelet transform extrema extrapolation**",** *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 4, pp. 2379 – 2382, 9-12 May 1995.

46. Chang, S. G., Z. Cvetkovic, M. Vetterli, M., "Locally adaptive wavelet-based image interpolation", *IEEE Transactions on Image Processing*, Vol. 15, Issue 6, pp. 1471–1485, June 2006.

47. Temizel, A., T. Vlachos, "Wavelet domain image resolution enhancement using cycle-spinning", *IEEE Electronics Letters*, Vol. 41, Issue 3, pp. 119–121, 3 Feb. 2005.

48. Temizel, A., T. Vlachos, “Wavelet domain image resolution enhancement”, *IEEE Proceedings on Vision, Image and Signal Processing*, Vol. 153, Issue 1, pp. 25–30, 9 Feb. 2006.

49. Mori, T., K. Kameyama, Y. Ohmiya, J. Lee and K. Toraichi, “Image Resolution Conversion Based on an Edge-Adaptive Interpolation Kernel”, *IEEE Pacific Rim Conference*, pp. 497–500, 22-24 Aug. 2007.

50. Carrato, S., G. Ramponi and S. Marsi, “A simple edge-sensitive image interpolation filter”, *Proceedings of International Conference on Image Processing*, Vol. 3, pp. 711–714, 16-19 Sept. 1996.

51. Nemirovsky, S. and M. Porat, “On Texture and Image Interpolation using Markov Models”, *Electrical Engineering Department, Technion – Israel Institute of Technology*, Haifa, 32000, Israel, 2008.

52. Kirshner, H. and M. Porat, “Are Polynomial Models Optimal for Image Interpolation”, *Proceedings of EUSIPCO 2008*, Aug. 2008.

53. Avcibas, I., B. Sankur and K. Sayood, “Statistical evaluation of image quality measures”, *Journal of Electronic Imaging*, Vol. 11, no. 2, pp. 206–23, Apr. 2002.

54. Wang, Z., A. C. Bovik, “A universal image quality index”, *IEEE Signal Processing Letters*, Vol. 9, Issue 3, pp. 81 – 84, Mar 2002.

55. Wang, Z., A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, “Image quality assessment: From error measurement to structural similarity”, *IEEE Transactions on Image Processing*, Vol. 13, Jan.2004.

56. Wang, Z., E. P. Simoncelli and A. C. Bovik, “Multi-scale structural similarity for image quality assessment”, *Proc. 37th IEEE Asilomar conference on Signals, Systems and Computers*, Vol. 2, Nov. 2002.

57. Wang, Z., L. Lu and A. C. Bovik, “Video quality assessment based on structural distortion measurement”, *Signal Processing: Image Communication*, Vol. 19, no. 2, pp. 121–132, Feb. 2004.

REFERENCES NOT CITED

Borman, S. and R. L. Stevenson, "Spatial resolution enhancement of low-resolution image sequences. A comprehensive review with directions for future research", *Lab. Image and Signal Analysis, University of Notre Dame, Tech. Rep.*, 1998.

Borman, S. and R. L. Stevenson, "Super-resolution from image sequences – A review", *in Proceedings of Midwest Symp. Circuits and Systems*, pp. 374-378, 1999.

Patti, A. J., A. M. Tekalp and M. I. Sezan, "A New Motion Compensated Reduced Order Model Kalman Filter for Space-Varying Restoration of Progressive and Interlaced Video", *IEEE Transactions on Image Processing*, Vol. 7, no. 4, pp. 543–554, Apr. 1998.

Oppenheim, A. V., R. W. Schafer and J. R. Buck, *Discrete-Time Signal Processing*, Prentice Hall Signal Processing Series, ISBN: 0-13-754920-2, 1999.

Wang, Y., J. Ostermann and Y. Q. Zhang, *Video Processing and Communications*, Prentice Hall Signal Processing Series, ISBN: 0-13-017547-1, 2002.