International Journal of Electronics and Communication Engineering.
ISSN 0974-2166 Volume 5, Number 2 (2012), pp. 219-234

http://www.irphouse.com

Projection of High Resolution Video Frames with Very Less Noise using 3-D Linear Projection

[1]M. Nagaraju Naik and [2] Dr. P. Rajesh Kumar

[1]Assoc. Prof., Dept. of ECE, MIST, Hyderabad, A.P., India
E-mail: nagraju.naik@gmail.com
[2]Assoc. Prof., Dept. of ECE, A.U. College of Engineering, Vishakhapatnam, A.P., India
E-mail: rajashauce@gmail.com

Abstract

The progressive video coding is demanding for high data rate and resource consumption for quality oriented data service. For the quality oriented video transmission in resource constraint channel, the method of lower resolution transmission and a low complexity, higher accuracy projection algorithm is proposed. The conventional approach of Fourier domain transformation for resolution projection is modified with the cubical spline projection and a 3-D projection approach to image interpolation for dyadic decimation is proposed. The evaluation for the suggested approach is presented an observed to be more effective comparative to linear interpolation.

Keywords: Image projection, cubical B-spline projection, 3-D projection, image coding, linear interpolation.

Introduction

Digital video codings are developed from a long time. Various methods have come up with various quality improvement approaches to achieve the objective of efficient video coding in real time systems. In the area of video processing there is a need to improve the resource requirement for progressive video processing using resource optimization techniques. In previous literature it is observed that video sequencing can be improved by optimized usage of available resources using scheduling schemes, multiplexing schemes, representation schemes etc. The proposed conventional methods were observed to be developed keeping available resources and there constrains in mind. With increase in demand for video coding various services were incorporated in video processing for representation, such as conferencing, live observations, navigations etc. All these applications demand high-resolution representation of video data for real time interfacing and communications. With the incorporation of developed optimization scheme as outlined above can provide a significant improvement in coding but in current scenario and for future applications these methods may get constrained. As the available resources such as bandwidth, power, coding techniques are limited to certain minimum values. As the demand for next generation technologies for incorporation of high standard video codings with accuracy, is coming up, there is a need to come up with distinct approaches in video coding for progressive transmission under constraint resources for image representation than the conventional approaches. To achieve high resolution representation videos are to be retained for good visual quality. For having good visual quality videos are to be processed in such a way that actual video information should not be effected. As resource optimizations are constrained, coding techniques are to be improved to achieve the stated quality in video processing. A new approach of video representation has emerged in recent past with high-resolution projection approach for low dimensional video sequence. Such method has a significance of low data processing, low resource requirement and lower complexity in computation compared to conventional coding techniques. Where data representation low is very low in resource requirement and has low visual quality then it is suggested to be interpolated on a high-resolution grid for better visual quality. To achieve higher visual quality the stated interpolation approaches were carried out in frequency representation using interpolation techniques. The most dominantly used interpolation is the fast Fourier based interpolation technique. It is observed that Fourier transformation transforms the video sequence from time to frequency with non-resolution description resulting in lower accuracy in video interpolation. During the interpolation of video sequence from low resolution to high resolution the projection is carried out in 2D projection, which doesn’t keep the frame sequence integrity resulting in poor visual quality. In this work the problem of low-resolution interpolation issue is focused to improve the resolution projection using cubic-b-spline interpolation with 3D-interpolation technique to develop a faster and accurate scaling approach compared to the Fourier based interpolation approach.

Past Works

The video High-resolution approach is observed to estimate a high-resolution (HR) version of a video sequence from its low-resolution (LR) equivalent. High-resolution representation is an open problem in video processing. Often there is insufficient amount of LR data resulting in formation of a various problems. Even in the case of a sufficient number of LR samples (i.e., more source samples than desired output samples), the presence of distortion and noise produces contradictory source information. For a solution to be possible, some assumptions have to be made regarding the nature of the relationship between the known LR and desired HR sequences. These assumptions can be made implicitly as a part of the reconstruction algorithm, such as a simple band limited model [1] or a specific regularization term [2], or can consist of a model estimated from the LR source data. Some recent examples of this latter case include various second-order statistical modeling techniques used in [3,4] both of which are concerned with the case of a single LR image. A power spectral density (PSD) modeling approach in [5], and use of support vector regression [6] is proposed for the projection approach. Since the precise relationship between the LR and HR sequences is unknown, there will be some error in the estimated sequence video sequence. Previously similar approaches have been proposed for video-to-video [8, 9] and video-to-image problems [10]. The basic solution is a relatively simple block-processing linear minimum mean-squared error (LMMSE) spatial-domain interpolation. In many ways the solution is similar to previous approaches for still-image scenarios, but certain modifications have been introduced to improve performance and tackle some complications that arise specifically for the case of video. There are several features for this proposed method. First, because it uses a local reconstruction, the required models can also be adjusted locally for a higher-quality HR reconstruction. The second major quality of the algorithm is its efficiency, a necessary requirement for a practical video enhancement solution. Finally, the proposed method offers significant flexibility in several aspects of the solution: the scaling factor can be arbitrarily adjusted (even separately on each dimension for aspect ratio conversion), the model information (distortion, noise, and motion) can be changed locally, and the amount of source data can vary allowing a more selective inter-frame registration. The solution approach is not intended for any single specific application and should prove useful in most resolution enhancing scenarios. Most previous approaches consider a still-image scenario, although many do not necessarily preclude video. A general overview of the problem is found in the survey paper [7]. The difficulties of the still-image problem are discussed in the many papers cited in this survey, as well as a large amount of subsequent work. Many of the regular still-image High-resolution techniques are easily modified for use in the video scenario. The primary complications unique to the video scenario are related to registration of data from a temporally varying source. In the still image case, the set of LR images can be assumed taken from a single unchanging scene. Registration differences are mainly due to differences in acquisition of device positioning. In contrast, for video the set of LR frames each represent a single capture moment of a temporally dynamic scene. This presents several complications: different objects can move separately over the frames, movement at different depths will cover and uncover portions of the scene (the occlusion problem), object features can change over time, etc. These complications require a registration algorithm, which can account for local motion. Perhaps the simplest solution is to use a standard block-based approach, which is easily incorporated with the motion information already available to standard codecs. An important distinction between block-based motion for registration and coding applications is that there is no error residual information available for High-resolution. Thus, there may exist significantly mismatched portions or contents, which can degrade High-resolution performance if not removed. There are alternatives to block-based registration, such as [11], which can provide a better representation of the natural motion found in video sequences. The proposed approach also includes data validation checks to improve robustness in the event of registration error. HR estimates, the computational requirements for the estimation are the matrix operations in [1] and [2], along with the computation required for estimating the necessary correlation matrices. While there exist many methods for determining correlation function estimates, the possibility of non-uniformly spaced data in x as well as the possibility (due to fractional scaling) of phase misalignments between the uniform LR and HR grids can complicate the estimation process. In simpler circumstances, the source data is uniformly spaced and the LR grid spacing is an integer multiple of the HR grid spacing. Previous methods (e.g., edge-directed interpolation [3] or PSD-based methods in [4,5] considered such circumstances, in which a uniform HR correlation function is estimated from its uniformly down-sampled equivalent. In parametric modeling approach for estimating image PSD [5], the approach is simplified for faster block processing. The correlation is found using a 2D fast Fourier transform (FFT) approach. Assuming an anti-aliasing distortion between the HR and LR versions of a block, the low frequencies of the HR block can be approximated as corresponding directly to the entire LR block. The high frequencies of the HR block are assumed lost in the anti-aliasing and are estimated. This approximation doesn't perfectly describe the true relationship, as non-ideal anti-aliasing filters still introduce some corruption of high frequency information into the LR representation. Though, this serves as a sufficient approximation, especially for the case of small blocks of data. An important observation, discussed in [5] and earlier works is that the lower frequencies of the spectrum as represented by a magnitude squared of the FFT totally dominate the higher frequencies, almost always by several orders of magnitude in standard images. Thus, replication of the complete LR spectrum onto the base-band of the HR spectrum replicates the vast majority of statistical information. This Fourier based estimation approach is observed to be lower in accuracy when projected on a higher grid plane during projection.

System Design

A 3-D scaling approach for video coding which provides a better performance at a lower bit-rate transmission is proposed. A Non-linear video enhancement compensation with the cubic-B-Spline filter is used to improve the decoded quality of the 3-D down-scaling for video coding as shown in Figure.1 The real-time communication scenario is illustrated in this figure, where this algorithm applies the 3-D linear decimation (3D-LI↓) as the encoder, and the 3-D linear interpolation (3D-LI↑) as the decoder for video coding. As a result, the proposed non-linear video enhancement compensation with cubic B-spline filter is used for the post-processing step of this decoder.

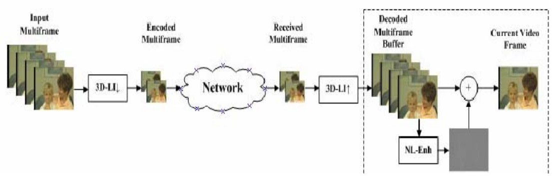

Figure 1: Proposed video coding approach

In this paper, we brief about the design approach of the proposed method, developed for projecting LR video values onto high-resolution grid based on conventional Fourier interpolation and cubic-b-spline approach. For the improvement of the projection scheme in this work a Cubic-B-Spline based 3D projection scheme is developed and the system approach developed is presented below,

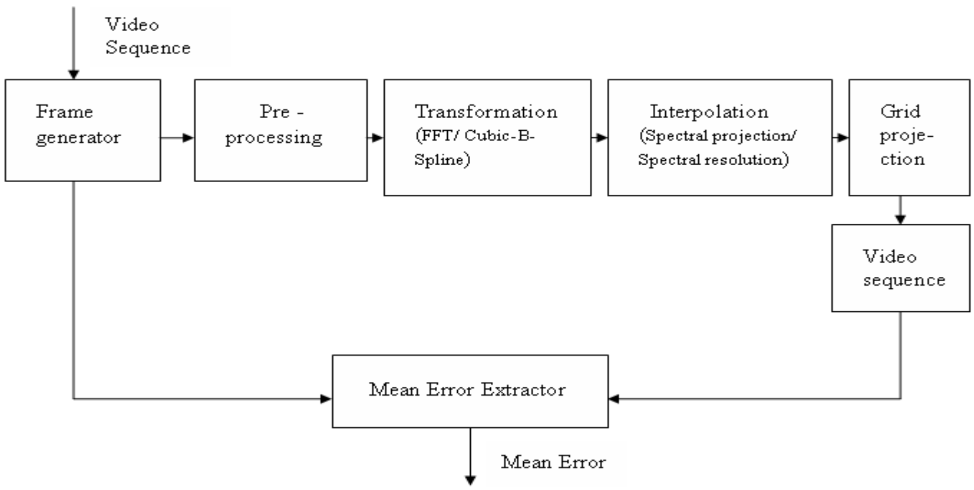

Figure 2: Architecture of proposed method

The Architecture of Proposed Method is shown in above, here frame generator takes low-resolution video sequences as input and converts it into static frames. These static frames converted into grey level in the pre-processing step. Next these grey level frames converted into frequency domain using FFT transformation to compare we are using Cubic-B-Spline method. The transformed data to than interpolated (spectral projection/spectral resolution) using FFT and Cubic-B-Spline. The projected data is aligned over a predefined grid format to obtain high-resolution video. This video sequence is compared with original data to extract Mean Error.

The above shown figure illustrates developed system architecture for the implementation of scalar projection of video stream using conventional and proposed scaling methods. The concept of resolution projection of video stream is developed using spectral and frequency interpolations and evaluated for computational time and retrieval accuracy.

Conventional Fourier Interpolation Technique

The Fourier algorithm is used to solve this problem. It concentrates mainly on fixing the low-frequency components, but it turns out that this algorithm also gives us some high-frequency components depending on the information in the LR videos. The two steps used to solve this problem are shown below. The details of each step is given in the following sections

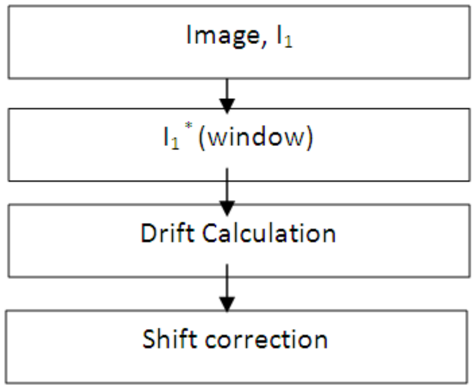

Figure 3: various operations done on the video before applying P-G Algorithm in step 1

The first step in any High-resolution algorithm is to estimate the motion between given LR frames. A good ME is a hard prerequisite for SR. In this project the motion is restricted to shifts and rotation, so a very simple (though accurate) approach is enough for video registration.

Transformation

There are two ways that can be used to carry out rotation estimation between low-resolution videos.

- Rotate individual videos at all the angles and correlate them with the first video. The angle that gives the maximum correlation is the angle of rotation between them.
- The method described above is computationally expensive, so a faster method is to calculate the video energy at each angle. This will give a vector containing average energy at each angle. The correlation of these energy vectors of different videos will give the angle of rotation. The energy calculation could also be carried in frequency domain.

$$\text{Angle}(i) = \max(\text{index}(\text{correlation } (I_1(\theta), Ii\ (\theta))))..\qquad (1.1)$$

Where $I_1(\theta)$ is the pixel intensity of the reference pixel and $Ii(\theta)$ is the intensity of the i^{th} pixel. It turned out that though the first method is computationally expensive, but gives more precise results, so it was used in this project.

Shift Calculation

Once rotation angle is known between different videos, shift calculation can be performed. Before calculating the shift, all the videos are rotated with respect to the first video. Block matching was not used here for two reasons. It is computationally expensive. We need sub-pixel accuracy that is very hard to get with block matching. A much faster and reliable way to achieve this is using frequency domain method.

For determining the amount of shift in any pixel of an video, we use the relation

$$Fi(u^T) = e^{j2\pi u\Delta s}.\ F_1\ (u^T)\ldots\ldots\ldots \quad (1.2)$$

This is obtained by applying Fourier Transform of a reference pixel matrix. The shift angle Δs from the above relation can be calculated as

$$\Delta s = [\text{angle}\ (Fi(u^T)/\ F_1(u^T))]/2\pi\ \ldots\ldots.. \quad (1.3)$$

and in matrix form, $\Delta s= [\Delta x\ \Delta y]^{T\ \cdots}$……… (1.1)Where
U(x, y) is the pixel coordinate

Δx is the variation of current x-position from reference x-position
Δy is the variation of current y-position from reference y-position
$Fi\ (u^T)$ is the transform of transposed i^{th} pixel
$F_1\ (u^T)$ is the transform of transposed reference pixel
Δs is the shift angle

Technically, each frequency domain method should give the same shift, but that does not happen. So, a least square method is used to estimate the shift Δs. As the LR videos could be aliased, only 6% of the lower frequencies were used to calculate Δs.

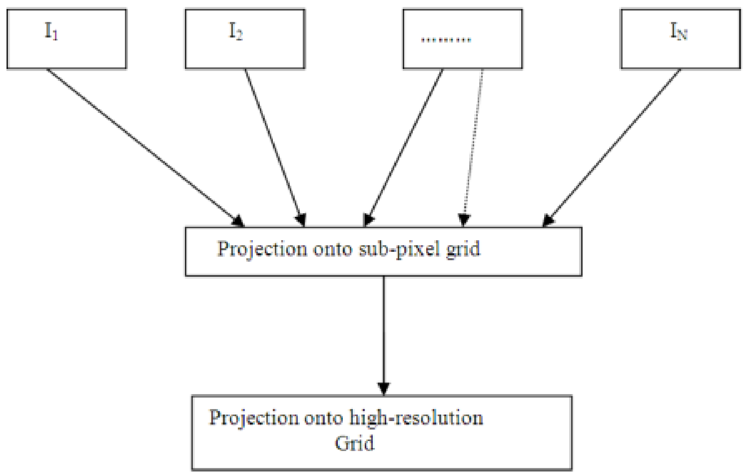

Figure 4: sub pixel grid mapping approach

Fourier Projection

Once we have a good estimate of shift and rotation between all the videos and the reference video, we can find the pixel positions of all the LR videos in the reference frame of the first video. Then we can project this information on high-resolution grid.

This method assumes two things:

- Some of the pixel values in the high-resolution grid are known.
- The high frequency components in the high-resolution video are zero.

It works by projecting HR grid data on the two sets described above. The steps are:

- Form a high-resolution grid. Set the known pixels values from the low-resolution videos (after converting their pixel position to the ref frame of first low-resolution video). The position on the HR grid is calculated by rounding the magnified pixel positions to nearest integer locations.
- Set the high-frequency components to zero in the frequency domain.
- Force the known pixel values in spatial domain. Iterate.

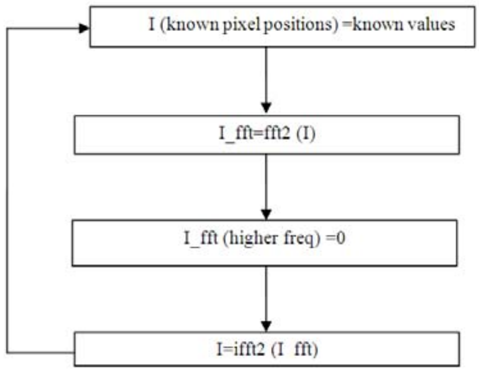

Figure 5: operational flow chart of the projection approach

By making the high-frequency equal to zero, this method tries to interpolate the unknown values and so correct the aliasing for low-frequency components. Also, by forcing the known values, it does predict some of the high-frequency values. The following set of videos walks through the actual working of this algorithm. Initially, we fill the HR grid with known pixel values and make the unknown pixel values to be zero. This can be observed in the video shown below. The vertical & horizontal black grid lines represent the unknown pixels. The DFT of this initial estimate shows both high and low-frequency components.

In the next step, the higher frequencies can be made zero in the frequency domain. This effectively is low-pass filtering the video. The center video shows the effect. The unknown pixels now have got some value, and the known values have gone down in amplitude, due to low-pass filtering. We can then increase the magnitude of known pixels by forcing them to what they should be (as can be observed in the rightmost video). This again creates some high-frequency components. So, basically by iteratively doing this again and again, we are correcting the low-frequency values (by guessing the values for unknown pixels) and finding some the high-frequency components by forcing the known values. After a few iterations, the HR video converges to an SR video, which can be observed above. By juggling between the two data sets, i.e. forcing the high frequency to zero and forcing the known values, we have estimated the value of unknown pixels. When the video is projected over a high resolution grid plane it is observed that due to the resolution projection using Fourier projection the distribution of the pixel in the grid plane goes erroneous as it is projected over the reference grid based on the energy distribution and it works under the assumption of reference grid pixel known. The method also Detroit in performance when the registered video is not the same of the original video. These limitation results in the blurriness or stretching of video pixel resulting in poorer visual quality. The method also doesn't consider any effect of the frame sequencing which results in further video quality deviation from the reference due to inter-frame sequencing. To over come the stated limitation in this work a 3 dimensional projection scheme with a non-linear filtration technique is developed which is briefed below.

Proposed 3D Cubic-B-Spline Approach

The scaling approach causes a blurred video because there is no power in the high-frequency component of enlarged video. To improve the quality of such a blurred video, the video enhancement is required for real time applications. Video Enhancement is very important topic in the researches. The principle of video enhancement is to process an video so that the result is more suitable than the original video in many applications. A typical video enhancement is achieved through the high-pass filter followed by the post-processing in order to make the video suitable. In other words, this method uses a typical principle behind un-sharp masking and high-boost filtering. Non-linear video enhancement is similar to the typical video enhancement except the high-pass filter is replaced by non-linear operations. This enhancement method uses the Gaussian-pyramid or filter subtracts and decimate (FSD) -pyramid representation of an video to extract the high-frequency component of original video as shown in Figure 6. The major non-linear step involves clipping and scaling the extracted components.

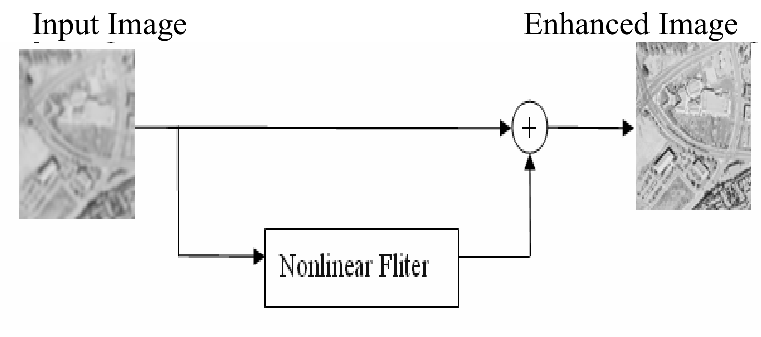

Figure 6: The proposed non-linear video enhancement

For the operation of the non-linear filtration an improved non-linear video enhancement that uses a cubic B-spline filter is proposed. In addition, a non-linear video enhancement compensation algorithm with 3D scaling approach is proposed to improve the quality of decoded video. The proposed enhancement algorithm applies the approach of the cubic B-spline filter to improve the non-linear video enhancement method. The Cubical spline approach is observed to provide higher in resolution information compared to FFT based resolution information. The approach of cubical spline approach is as defined below, for equally spaced sampled data $f(x_k)$ many interpolation functions can be written in the form,

$$\hat{f}(x_k) = \sum_k c(k)\beta(x - x_k) \ldots\ldots\ldots\ldots \quad (1.5)$$

Where $\hat{f}(x_k)$ is the corresponding interpolation function, β (x) is the interpolation kernel, and x and X_k represent continuous and discrete values, respectively. c(k) is the interpolation coefficient that depends on the sampled data $f(x_k)$. If a kernel satisfies,

$$\begin{cases} \beta(0) = 1 \\ \beta(x) = 0. |x| = 1,2,... \end{cases} \ldots\ldots\ldots\ldots\ldots\ldots \quad (1.6)$$

Then it can avoid smoothing and preserve high frequencies. They are called interpolators. Traditionally, we set c(k)=f(k) that's because we assume that $\beta(x)$ equals unity at the origin, namely, it satisfies equation (1.6). By doing this we make sure that for any integer k, the following equation holds true:

$$\hat{f}(k) = f(k) \ldots.. \quad (1.7)$$

Cubic B-spline kernel is given by

$$\beta(x) = \begin{cases} \frac{2}{3} - \frac{1}{2}|x|^2 (2 - |x|), & 0 \le |x < 1| \\ \frac{1}{6}(2 - |x|)^3, & 1 \le |x| < 2 \\ 0, & elsewhere \end{cases} \ldots\ldots. \quad (1.8)$$

Apparently, cubic B-spline kernel is not an interpolator, since, $\beta(0) = 2/3$ and $\beta(-1) = \beta(1) = 1/6$. Since cubic B-spline kernel differs from unity at the origin, we must set $c(k) \neq f(k)$ to make equation (1.7) still holds true. Let $f(x_k)$ be the available data, and $\hat{f}(x_k)$ be the value to be interpolated. Suppose that its nearest neighbors are located at coordinates x_k and x_{k+1}, and the spacing of the sampling grid be one for these data. We define:

$$s = x - x_k, \ \ldots\ldots\ldots\ldots \quad (1.9)$$

$$1 - s = x_{k+1} - x, \ \ldots\ldots\ldots\ldots\ldots \quad (1.10)$$

Where $0 \le s \le 1$ and $x_k \le x \le x_{k+1}$

For a discrete signal {f(k)} defined on $k = -\infty,....,+\infty$ Such that f (k) can be represented as

$$f(k) = \sum_{i=-\infty}^{+\infty} c(i) b_l^n (k-i) \ldots\ldots.,.., \qquad (1.12).$$

This expression can also be written as

$$f(k) = b_l^n * c(k) \ldots\ldots\ldots.. \qquad (1.13)$$

Where $b_l^n(k)$ is the finite impulse response of the operator that is referred to as the indirect spline filter of order n.

Cubic-B-Spline Filter

We can write equation (1.14) in the following form

$$\hat{f}(x) = \sum_{l=0}^{3} s^l v(l) \ldots\ldots\ldots\ldots.. \qquad (1.14)$$

Where

$$v(l) = \sum_{i=-2}^{l} b_l(i) c(k-i) \ldots\ldots \qquad (1.15)$$

Equation (1.11) is a polynomial in s. Nested evaluation of (1.11) is the most efficient approach. So equation (1.11) can be written as

$$\hat{f}(x) = \{[v(3)s + v(2)]s + v(1)\}s + v(0) \ldots \qquad (1.16)$$

In other words, the 2-D filter of cubic B-spline function is given by, [1, 1, 1; 1, 16, 1; 1, 1, 1] / 36.

For the obtained cubic-b-spline filter the developed approach is applied to improve the resolution performance as briefed below,

The low-frequency video I_1 is obtained from the input-blurred video I_0 using the cubic B-spline filter, and the high-frequency video Ko , called the residual video, is obtained by subtracting the low-frequency video I_1 from the input-blurred video I_0 , i.e., $K_0 = I_0 - I_1$. The enhanced video I_{-1} is generated as the sum of the input blurred video I_0 and the predicted higher-frequency video K_{-1} that is,

$$I_{-1} = I_0 + K_{-1} \ldots\ldots\ldots.... \qquad (1.17)$$

Where $K_{-1} = NL(K_0)$ is a non-linear operator of K0 , which includes both scaling and clipping steps, defined as follows:

$$NL(K_0) = s \times Clip(K_0)\,(1_0)\ldots \qquad (1.18)$$

Where the scaling constant ‘s’ is ranging between 1 and 10 and Clip(x) is given by

$$Clip(x) = \begin{cases} T, & if \quad x > T \\ x, & if \; -T \le x \le T \\ -T, & if \quad x < -T \end{cases} \ldots\ldots \qquad (1.19)$$

Where x is the pixel of the high-frequency video K_0 , $T = c \times K_{omax}$, K_{omax} is the maximum pixel of the high-frequency video K_0 and the clipping constant 'c' is ranging between 0 and 1. After a non-linear operator, the higher-frequency video K_{-1} can be utilized to enhance the input-blurred video I_0 .

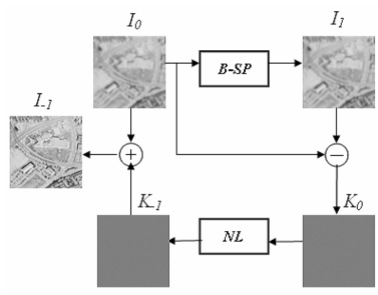

Figure 7: The proposed non-linear video enhancement method

In order to obtain a very low bit-rate video, a new type of three-dimensional (3-D) down-scaling scheme is presented for video coding. This scheme applies a 3-D decimation with a compression ratio of 8 to 1 as the pre-processing step of the encoder. As a consequence, a 3-D interpolation with a ratio of 1 to 8 is used for the post-processing step of the decoder.

Let t_1 ,t_2 , and t_3 be the integer indices and n_1 , n_2 and n_3 are also integers. The 3-D decimated scheme takes an video X (t_1 ,t_2 , t_3) as an input and produces an output of Y (t_1 ,t_2 , t_3) by a factor of 2 in each dimension as follows:

$$Y(t_1,t_2,t_3) = avg\left(\sum_{i_1=0}^{1}\sum_{i_2=0}^{1}\sum_{i_3=0}^{1} X(2t_1+i_1, 2t_2+i_2, 2t_3+i_3)\right)\cdots \quad (1.20)$$

for $0 \le t_i \le n_i - 1, \quad i = 1,2,3.$

Where avg(.) is returns the average (arithmetic mean) of a set of numeric values. Figure 8 shows the down-sampling of pre-processing stage using the 3-D linear method.

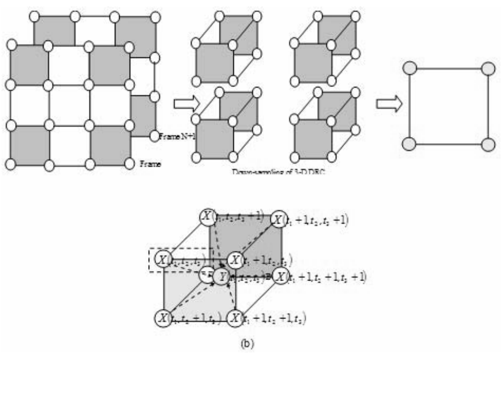

Figure 8: A 3-D linear decimated scheme

Using the decimated video Y(t_1 ,t_2 , t_3) obtain by equ.(1.20), the 3-D reconstructed video can be calculated by a linear interpolation shown in Figure 9 and given by

$$\hat{X}(t_1,t_2,t_3)=\sum_{k_1=0}^{1}\sum_{k_2=0}^{1}\sum_{k_3=0}^{1}Y(k_1,k_2,k_3)R(t_1-2k_1,t_2-2k_2,t_3-2k_3)\cdots \quad (1.21)$$

For $0 \le t_i \le 3, \quad i = 1,2,3.$

Where $R(t_1 - 2k_1, t_2 - 2k_2, t_3 - 2k_3)$ is the 3-D linear functions defined by

$$R(t_1,t_2,t_3) = R(t_1)R(t_2)R(t_3) \ldots\ldots \quad (1.22)$$

and R(t) is the 1-D linear function given by

$$\begin{aligned} R(t) &= \{1 - |t|/2 \quad ,|t|/2 \\ R(t) &= 0 \quad ,otherwise \end{aligned} \quad \ldots\ldots\ldots \quad (1.23)$$

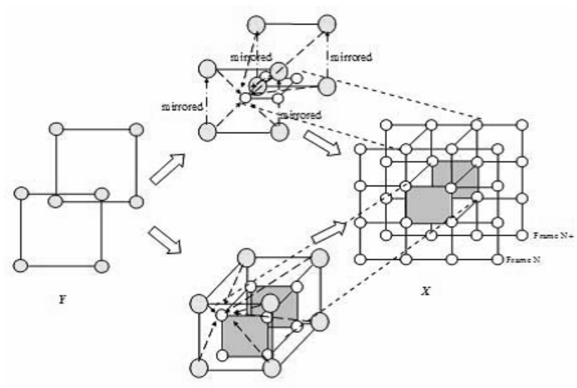

Figure 9: A 3-D linear interpolated scheme

The scaling operation is carried out for the given video sequence where the scaling operation is defined as, Let x_s, y_s be the size in the X direction and Y direction respectively of a planar source image. Let the corresponding sizes of the destination image be x_d, y_d. This means that the source image has x_s pixels in a row, which must be converted to x_d pixels in the destination image. The ratio of these two numbers is called scaling factor

$$S_x = \frac{x_d}{x_s} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (1.21)$$

Similarly the scaling factor in the Y direction is

$$S_y = \frac{y_d}{y_s} \ldots\ldots\ldots\ldots\ldots\ldots.. \quad (1.25)$$

Note that the scaling factor is the ratio of two integers and is hence a real (floating-point) number. We will just illustrate the method for scaling in the X direction and the same kind of operations can be duplicated for Y direction. After projecting low-resolution frames on high-resolution grid by using Cubic-B-Spline and conventional Fast Fourier Transforms methods, these high-resolution grids converted into the high-resolution video. Then mean of a resulted high-resolution video is compared with the mean of the original video, for calculate visual clarity and accuracy of the methods. The mean of a video frame/video is calculated by

$$Mean = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} X_{i,j}}{M * N} \quad \ldots\ldots\ldots\ldots \quad (1.26)$$

Which method gives the nearest or equal Mean value compare to original video Mean is the best method. Among FFT and Cubic-B-Spline methods, Cubic –B-Spline is the best method. For the evaluation of the suggested approach the simulation operation is carried out as presented.

Result Observation

Sequences of image frames are taken for performing simulation observation. A low dimension frame sequence representation is taken and is processed for different scale rates. The coding bit depth is also varied and evaluated. The observation obtained for the simulation is as outlined below,

Figure10: Original LR sequence of image frames

Figure 11: projected HR sequence of the processed image

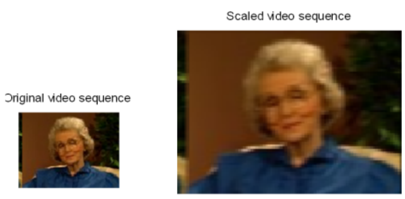

Figure 12: scaled sequence in 1:2 resolution using linear interpolation

Figure 13: interpolated sequence using cubic B-spline with 3D projection

Figure 14: interpolated image at 1:2 ratio using proposed approach

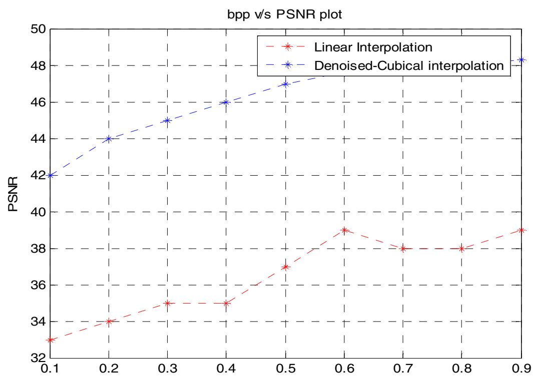

Figure 15: PSNR observation of the 2 methods

Figure illustrates the obtained PSNR observations at various rate of pixel coding. it is observed that where linear interpolation based approach achieves about 32 to 40 dB the proposed approach reaches about 42 to 50 dB at different coding bpp.

Conclusion

The energy spectral resolution projecting is carried out using Fourier transform techniques, where a low dimensional video sequence is projected to a high grid based on energy distribution. To improve resolution accuracy, a frequency based projection scheme is developed. To realize the frequency spectral resolution Cubic-B-Spline method is used. It is observed that the resolution accuracy with respect to visual quality, mean error and computational time is comparatively improved compared to conventional Fourier based interpolation technique. For the evaluation of the suggested approach, the system is tested over various low dimensions of video sequence and scaled over fixed and fractional scaling value. It is observed that the incorporation of frequency spectral information to energy spectral could provide better accuracy in resolution projection to high scaling value than conventional approach.

References

[1] S. P. Kim, N. K. Bose, and H. M. Valenzuela, "Recursive reconstruction of high resolution video from noisy undersampled multiframes", IEEE Trans. Acoust., Speech, Signal Processing, vol. 38, pp. 1013-1027, June 1990.

[2] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar, "Fast and robust multiframe High resolution", IEEE Trans. Video Processing, vol. 13, pp. 1327-1344, Oct. 2004.

[3] X. Li and M. T. Orchard, "New edge-directed interpolation", IEEE Trans. Video Proc., vol. 10, pp. 1521-1527, Oct. 2001.

[4] H. A. Aly and E. Dubois, "Specification of the observation model for regularized video up-sampling," IEEE Trans. Video Processing, vol. 14, pp. 567-576, May 2005.

[5] R. S. Prendergast and T. Q. Nguyen, "Spectral modelling and Fourier domain recovery of high-resolution videos from jointly undersampled video sets", under review for IEEE Trans. Video Proc., submitted Dec. 18, 2006.

[6] K. S. Ni and T. Q. Nguyen,"Video Highresolution using support vector regression", IEEE Trans. Video Proc., vol. 16, pp. 1596- 1610, June 2007.

[7] S. C. Park, M. K. Park, and M. G. Kang, "High-resolution video reconstruction: a technical overview", IEEE Signal Processing Mag., vol. 20, pp. 21-36, May 2003.

[8] B. Narayanan, R. C. Hardie, K. E. Barner, and M. Shao, "A computationally efficient High-resolution algorithm for video processing using partition filters," IEEE Trans. on Circ. Syst. For Video Technology, vol. 17, no. 5, pp. 621-634, May 2007.

[9] S. Farsiu, M. Elad, and P. Milanfar, "Video-to-video dynamic Highresolution for grayscale and color sequences," EURASIP Journal of Applied Signal Processing, Special Issue on Highresolution Imaging, vol. 2006, pp. 1–15, 2006.

[10] R. C. Hardie, "A fast video High-resolution algorithm using an adaptive Wiener filter," IEEE Trans. Video Proc., vol. 16, no. 12, pp. 2953-2964, Dec. 2007.

[11] H.-M. Chen, C.-Y. Hsieh, and G. Liao, "Non-rigid video registration using adaptive grid generation: preliminary results", in IEEE Conference on Medical Imaging, Washington DC, pp. 580-583, April 2007.