High- Resolution Video Scaling using Cubic-B-Spline Approach

T.Narsimulu
M.Tech
Hyderabad

B.Rama Raj
Asistant professor
C.M.R.C.E.T.
Hyderabad-501401.

Ch.Rajitha Laxmi
Asistant professor
K.G.Reddy College of Engg &Tech.
Moinabad.

ABSTRACT

Data coding have emerged in different directions. Various real time applications focused towards the improvement of data codings for efficient and accurate usage of available resources. In such application where resources are constrained, data coding systems are facing the limitation of processing speed and accuracy. Applications where speed of computation for transferring is more important, accuracy is neglected or vice versa. In such systems efficiency could be improved with a low computation, lower dimension coding techniques with higher accuracy. One such application where these effects are predominantly been observed is “video coding”.

Video coding has limitations of large data processing resulting in slower or low accurate systems. To improve the performance of such systems various approaches were made. These approaches are limited when spatial resolution variation, were dominantly observed. It is observed in past literatures that lower dimensional image representation could reduces the resource requirement, but are found limited with accuracy when displayed in real time. To achieve higher accuracy in displaying high-resolution projection, various approaches using video scaling methods were seen. These projection methods were found to be more accurate in displaying, when processed in frequency representation.

To transform an input data from time representation to frequency representation, conventional Fourier transformation technique were used. Fourier transformations are limited in representations and interpolation for high resolution varying video sequences. Additionally this approach is not effective much in interpolation of video sequence when represented in a low dimensional representation. In this work a focus is made to improve the accuracy from low representing video sequence to high-scaled video output using advance interpolation technique called ”Cubic-B-Spline” approach. The suggested method is developed on MATLAB tool with image and video processing toolboxes to implement and evaluate the suggested method for system accuracy and efficiency.

1. INTRODUCTION

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 low data representation 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.

2. VIDEO SCALING

Unlike most other information technologies, which have up come an exponential growth for the past several decades, display resolution has largely constrained. Low display resolution has limited the resolution of digital videos. Now that large-scale, high-resolution displays have become available, it opens up opportunities for scalable high-resolution digital video delivery. Image/video scaling is the process of resizing a digital image. Scaling is a non-trivial process that involves a trade-off between efficiency, smoothness and sharpness. As the size of an image is increased, so the pixels, which comprise the image, become increasingly visible, making the image to appear soft. Conversely, reducing an image will tend to enhance its smoothness and apparent sharpness.

Up-scaling video players contain a video scaler, which allows the user to convert lower resolution content into a signal that the display device will handle as high definition content. Depending on the quality of the scaling that is done within the Up-scaling [2] video player, the resultant output quality of the video displayed may or not be improved. The idea behind Up-scaling/Up-converting video players is that when a video players connected to an HDTV, especially one of the fixed pixel display type such as LCD, Plasma display, or DLP and Lcos projection TV, scaling happens anyway, either inside the player or the TV. By performing the scaling closer to the source inside the VIDEO player, the video scaler gets to work with the original signal without the concern of transmission error or interference. There exist independent benchmark tests verifying that some up converting video players do produce better video quality. Video information can only be retained or lost in each successive conversion step, but not created.

The technology used in hybrid systems may soon be found in resource constrained systems, where constraints in form factor, battery life, and processing power only allow video capture at much lower spatial resolution and HR stills[3]. Video-resizing methods that use HR stills, as guides would therefore be very useful for producing HR video or enlarging particular frames. Current approaches use High-resolution methods such as optical flow-based framework, Example-based and robust correspondence-based methods.

Most of these algorithms have high computational complexity and may be unsuitable for real-time and embedded systems such as camera device. The researching approaches for resizing video using HR stills were developed which implicitly use a layered approach to fill in details using HR stills. Focus on developing technologies that resize images and video are made. Various resizing approach in applications like mobile phones, where images captured at low resolution (e.g., 1MB) would appear as if they were taken at much higher resolution (e.g., 5MB). These methods could also be used in the digital-zoom features employed in low-end camera phones. Resizing and quality enhancement of compressed images and video are important directions for further research.

3. VIDEO SCALAR

Video Scalar[4]is a device that stretches or shrinks images or video frames. For the purposes of illustration, for a source image as shown below:

When expand the size of the image, the total number of pixels contained in the destination image is more than what is available in the source image. Hence a whole lot of new pixels have to be created by the scaler.

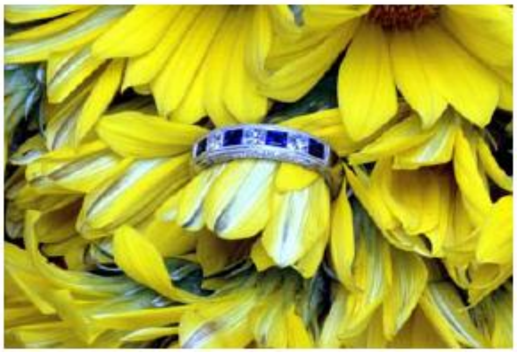

Figure 3.1 Source Image

When shrink the image, a part of the pixels are kept and remaining are discarded. The scaling systems job is outlined as: (1) Decides what pixels to keep in the source image and (2) what pixels have to be recreated.

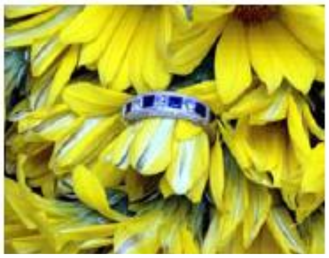

Figure 3.2 shrinked Image

A scalar can stretch an image only in X direction or only in Y direction or both. A scalar that can achieve arbitrary image sizes without noticeable quality loss is to be developed. There is no restriction to keep the aspect ratios constant while scaling. A high-resolution image representation for the shrinked image is shown in figure 3.3 below.

No distinction is made between shrinking and expansion of an image as far as the scalar algorithm[5] is concerned. They are the opposite sides of the same problem. The scalar will operate exactly the same way if we are expanding or squeezing the source image. The available scalar unit process with smooth expansion/contraction scaling. The construction of a hardware system, which can operate at video frame rates[6] are also focusive. In particular, the focus of the system is to develop different parameters that control the performance of the scalar.

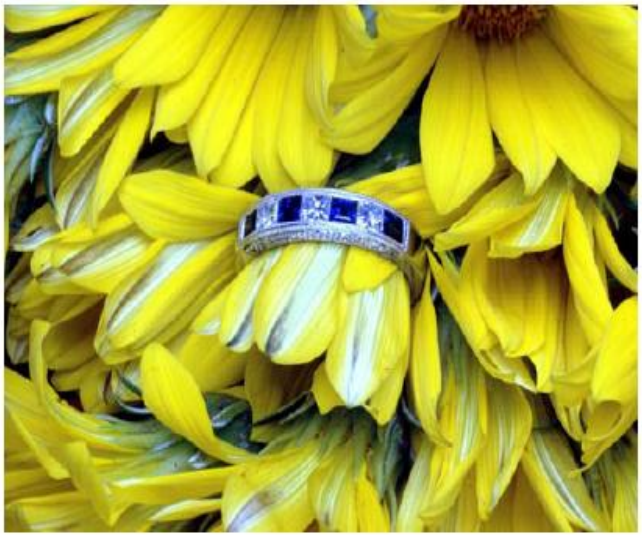

Figure 3.3 Expanded Image from shrinked Image

4. DESIGN APPROACH

4.1 Introduction

The problem of video high-resolution is observed to estimate a high resolution (HR) version of a video sequence from its low-resolution (LR) equivalent. 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, the presence of distortion and noise produces corrupted video information. In many ways the solution is similar to previous approaches for still-video scenarios, but certain modifications have been introduced to improve performance and tackle some complications that arise specifically in the case of video[7].

There are several attractive features for this proposed method. First, 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 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[8].

4.2 System architecture:

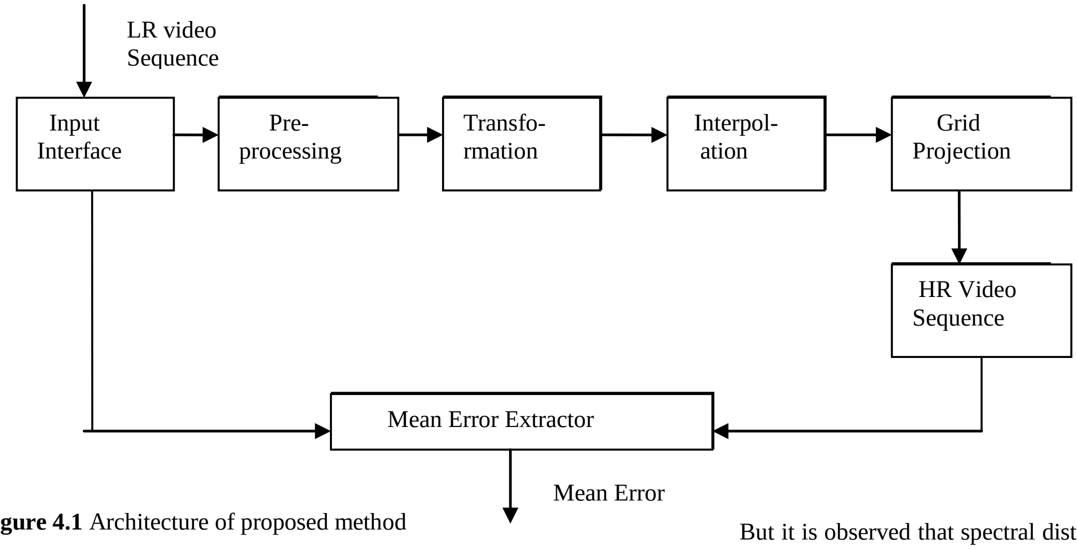

Figure 4.1 Architecture of proposed method

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[9]. 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.

4.3. Functional Description:

4.3.1. Input Interface

The developed system is processed over a very low video sequence represented in low dimensional projection. To evaluate the performance of suggested scaling system, a low dimensional, colored video streams are read and transformed into frame sequence using input interface unit. The processed frame sequence is then passed to a pre-processing unit for the equalization of input frame sequence for further processing.

4.3.2. Pre-Processing

This unit extracts the gray pixel intensity of the continuous frame sequence and pass to the interpolation unit for further processing. The gray pixel intensity are extracted from the input information segregated colored information.

4.3.3. Transformation Unit

This unit transforms the given input information into power spectral distribution using Fourier interpolation. It is observed in conventional architectures that the energy distribution of the original data could be used as interpolating information to represent high quality videos.

But it is observed that spectral distributions need not be sufficient for accurate interpolation, as the frequency resolution for spectrum energy coefficients may vary distinctly. To achieve better representation Cubic-B-Spline method is incorporated for such requirement.

4.3.4. Interpolation

Once the spectral resolutions were obtained, the pixel is to project on a higher grid level depending upon the scale value. Scaling of the video is achieved by interpolating the pixel information based on energy distribution of the given video sequence. To achieve better interpolation rather than energy resolution, spectral resolution could provide high-resolution accuracy developed using Cubic-B-Spline approach. The interpolated information[10] is than projected on a grid projection to represent the given low dimensional video sequence into high-resolution video sequence.

4.4. PROPOSED 3D CUBIC- B-SPLINE

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[1]. 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.4.2. The major non-linear step involves clipping and scaling the extracted components [2].

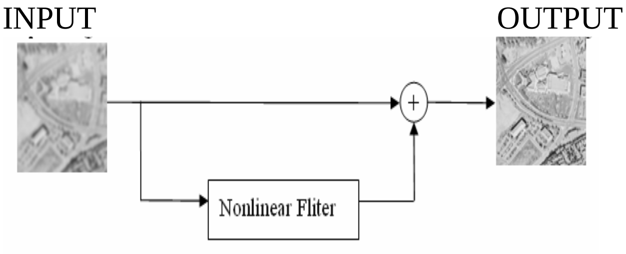

Figure.4.2. The proposed non-linear video enhancement

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.

4.5. Proposed Enhancement Scheme

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[1,2],

For equally spaced sampled data $f(x_k)$ many interpolation functions can be written in
the form

$$\hat{f}(x_k)=\sum_k c(k)\beta(x-x_k) \text{ ………………… (4.1)}$$

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

$$\begin{cases}\beta(0)=1\\ \beta(x)=0.|x|=1,2,\ldots\end{cases} \text{ ……………… (4.2)}$$

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 (4.6). By doing this we make sure that for any integer k, the following equation holds true:

$$\hat{f}(k)=f(k) \text{ …………………….... (4.3)}$$

Cubic B-Spline kernel is given by

$$\beta(x)=\begin{cases}\frac{2}{3}-\frac{1}{2}|x|^2(2-|x|), & 0\le|x<1| \\ \frac{1}{6}(2-|x|)^3, & 1\le|x|<2 \\ 0, & elsewhere \end{cases} \text{ ……….(4.4)}$$

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 (4.7) 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:

$$s=x-x_k, \text{ ……………….……(4.5)}$$

$$1-s=x_{k+1}-x, \text{ ……………..……..…(4.6)}$$

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

$$\begin{aligned} f(x)=&\frac{1}{6}(1-3s+3s^2-s^3)c(k-1)\\ &+(\frac{2}{3}-s^2+\frac{1}{2}s^3)c(k)\\ &+(\frac{1}{6}+\frac{1}{2}s+\frac{1}{2}s^2-\frac{1}{2}s^3)c(k+1)\\ &+\frac{1}{6}s^3c(k+2)\end{aligned} \text{ …(4.7)}$$

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

$$f(k) = \sum_{i=-\infty}^{+\infty} c(i) b_l^n (k-i) \quad \text{………………….,(4.8).}$$

This expression can also be written as

$$f(k) = b_l^n * c(k) \quad \text{……………...…..…(4.9)}$$

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.

4.6. Proposed 3-D Down scaling for video coding:

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[11].

4.6.1. Linear decimated scheme:

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:

$$Y(t_1, t_2, t_3) = \quad avg\left(\sum_{i\ =0}^{1} \sum_{i_2=0}^{1} \sum_{i=0_3}^{1} X(2t_1 + i_1, 2t_2 + i_2, 2t_3 + i_3) \right)$$

………(4.20)

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

Where *avg*(.) is returns the average (arithmetic mean) of a set of numeric values. Figure.4.3 shows the down-sampling of pre-processing stage using the 3-D linear method[13,14].

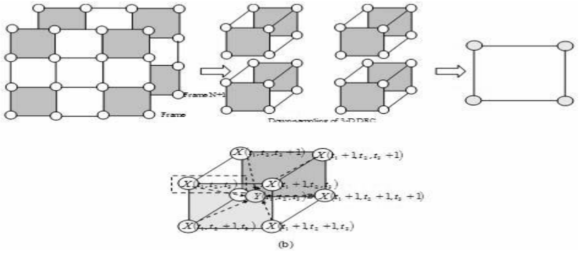

Figure.4.3. A 3-D linear decimated scheme

4.6.2. Linear interpolated scheme:

Using the decimated video Y(t$_1$,*t$_2$* , *t$_3$*) obtain by equ.(4.20), the 3-D reconstructed video can be calculated by a linear interpolation[15] shown in Figure.4.4. and given by

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

(4.21) 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

$$R(t_1, t_2, t_3) = R(t_1) R(t_2) R(t_3) \quad \text{…………(4.22)}$$

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

$$R(t) = 1 - |t|/2 \quad , |t|/2$$
$$R(t) = 0 \quad , otherwise \quad \text{………….………(4.23)}$$

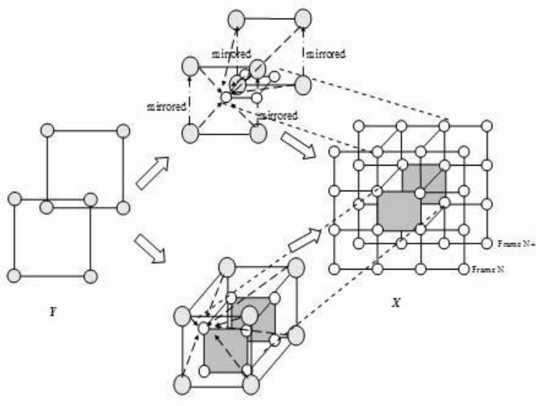

Figure.4.4. A 3-D linear interpolated scheme

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

$$S_x = \frac{x_d}{x_s} \quad \text{…………………..…. (4.24)}$$

Similarly the scaling factor in the Y direction is

$$S_y = \frac{y_d}{y_s} \text{………………..… (4.25)}$$

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

$$Mean = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} X_{i,j}}{M * N} \text{…………………(4.26)}$$

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. The corresponding results and comparisons of conventional Fast Fourier Transform and proposed method are discussed in the next section.

5. RESULTS AND OBSERVATIONS

5.1 Input consideration:

The suggested system is developed for high-resolution video projection using 3D Cubic-B-Spline (3D-CBS) scaling approach based on energy spectral and frequency spectral resolution representation. For the evaluating of the suggested approach in this work a continuous video sequence with various minimum dimensions are processed and evaluated over different scaling factors. The computational time revealing the complexity of the approach is evaluated. The system accuracy with respect to mean error between input frame and output is calculated. The obtained simulation observations are as outlined below.

5.2 Simulation observations

Observation 1:

Sample: Grandma.avi

Size: 118Kb

Resolution: 72x92

Color: RGB color plane

Frame rate: 30 frame/second

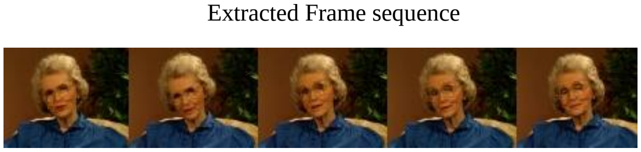

Figure 5.1: Captured frame sequence of the grandma video file

Figure above illustrates the original frame sequence taken for processing, from captured video sequence of grandma. The original frame sequence is taken at a low resolution with pixel representation of 72x92per frame size. These 5 frame sequences are passed to the developed system for pre processing.

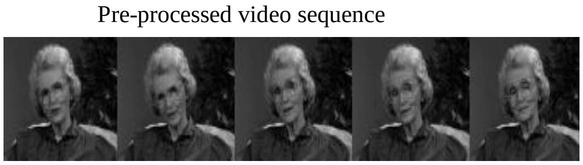

Figure 5.2 Pre-Processed video sequence of grandma. The preprocessed sample is as shown above. The frames are transformed to gray level so as to process on the pixel values before computation.

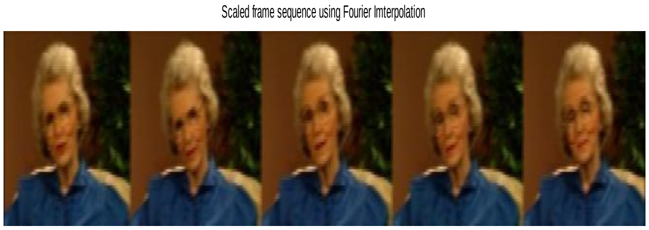

Figure 5.3 FFT based scaled frame sequence at a scale factor of 1:2 for the given test
Sample

These frames are read frame by frame to the processing unit to project it into a higher resolution. The projection is carried out at 1:2 scale levels and the retrieved observation is as shown below.

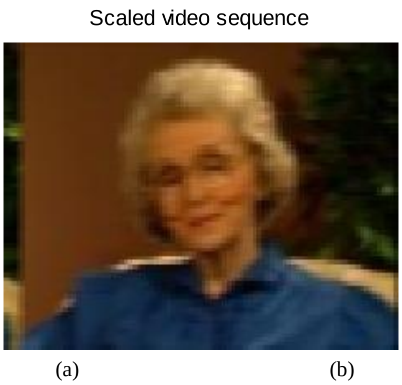

Figure 5.4: (a) A LR video sequence of grandma at 72x92 dimension used for testing,
(b) Obtained scaled video sample at 1:2 ratio using FFT-interpolator

Above figure shows the original video sequence of the grandma video sample consisting of 5 frames in color frame. The obtained video sequence at a scale of 1:2 ratio using FFT-interpolation is shown in figure 5.4(b).

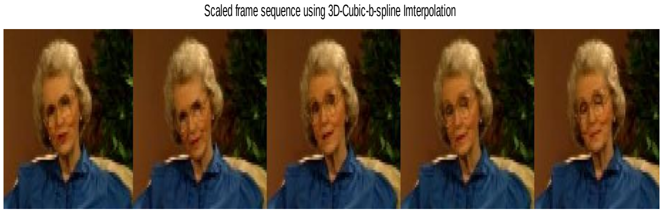

Figure 5.5: Scaled frame sequence obtained using 3D-CBS approach at 1:2 ratios

The frames are transformed to process on the pixel values before computation. These frames are read frame by frame to the processing unit to project it into a higher resolution. The projection is carried out at 3D-CBS approach and the retrieved observation is as shown below.

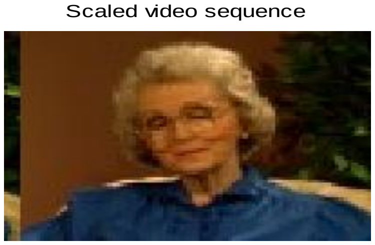

Figure 5.6: Scaled video sequence using 3D-CBS approach at 1:2 ratio

Above figure shows scaled video sequence obtained using 3D-CBS of 1:2 ratio frame, the obtained video quality will be better then the FFT-interpolation approach

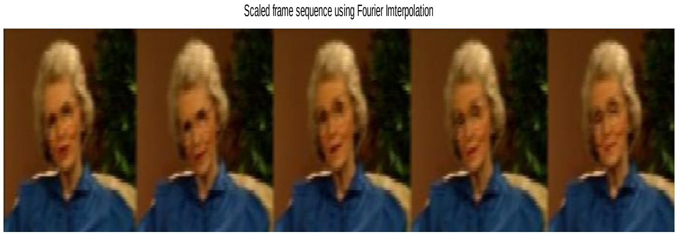

Figure 5.7: Scaled video sequence at scale factor 1:3.2 rate using FFT-interpolation

The retrieved scaled video after the projection of video using Fourier transform is as shown in figure 5.7. The interpolation is carried out for the spectral distributed video coefficients obtained after Fourier interpolation. The interpolation is made for the spectral distributed data as shown above.

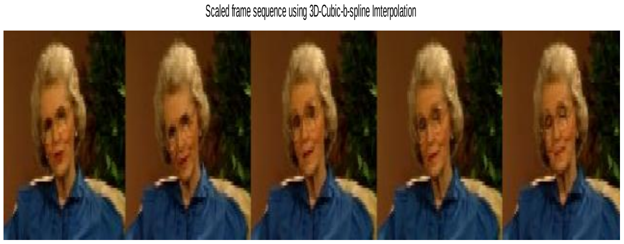

Figure 5.8: Scaled video sequence at scale factor 1:3.2 rate using 3D-CBS interpolation

The obtained result observation for a video scaling operation after the cubical b-spline approach is as shown in figure 5.8. The observation clearly illustrates the accuracy in retrieval in terms of visual quality as compared to the conventional Fourier based coding technique. The scaling operation of 1:3.2 ratio is carried out for the given low resolution video sequence.

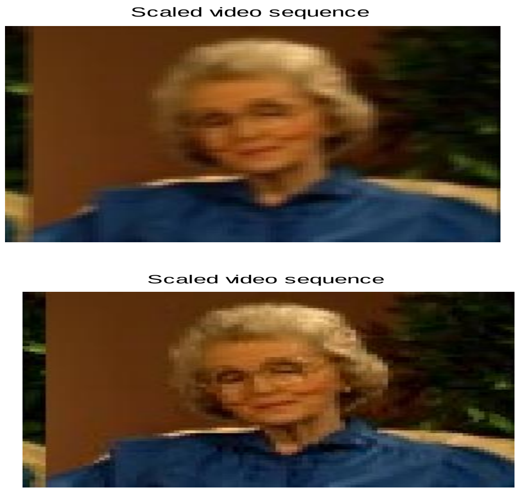

Figure 5.9: (a) Scaled image using FFT-interpolation at 1:3.2 ratio

(b) Scaled image using 3D-CBS interpolation at 1:3.2 ratio

Above figure illustrate the comparison of FFT-interpolation approach and 3D-CBS interpolation approach. Figure 5.9(b) 3D-CBS interpolation scaled video sequence having better quality then the Figure 5.9(a) FFT-interpolation.

5.3. PERFORMANCE and OBSERVATION

The below above displays comparison of conventional FFT method with proposed Cubic-B-Spline method with respect to Computation time and Mean Error(ME) for different scaling factors. From the above table it is clearly observed that proposed method performs better than conventional FFT methods because proposed (Cubic-B-Spline) method takes less computation time and gives less Mean Error compared to conventional FFT method for different scaling ratios.

Table 5.1 Comparison table between FFT and Cubic-B-Spline methods

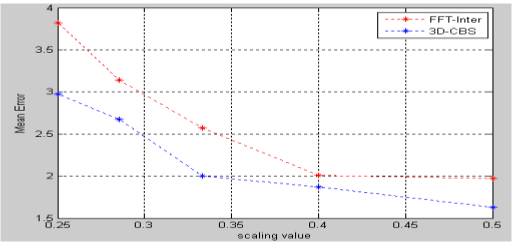

Figure 5.10 Mean Error comparison plot for the two methods

As a quality factor for the measurement of the system accuracy in projection the mean error between to mean values of the original video and the projected video is carried out. The distance of the mean values from the two original videos illustrates the accuracy of retrieval for the two methods.

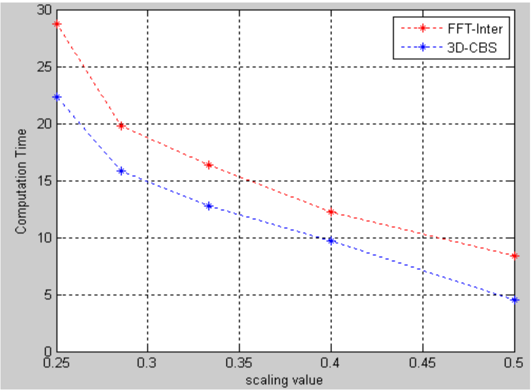

Figure 5.11 Computation time taken for the two methods

The system developed is also evaluated for the computation time taken for the computation and projection of the frame sequence for interpolation. The total time taken for reading, processing and projecting is considered for the processing system.

6. CONCLUSION

In this work a high scaler video projection approach is developed. High scaler representation of video sequence is limited due to video information present in low dimensional video sequence. To project a video frame sequence into high-resolution static or fractional scaling value, a scaling approach is developed based on energy spectral interpolation and frequency spectral interpolation techniques.

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.

Scale Value	Computation Time		ME	
	FFT-Inter	3D-CBS	FFT-Inter	3D-CBS
1:2	8.376	4.532	1.97	1.63
1:2.5	12.247	9.731	2.01	1.87
1:3	16.321	12.760	2.57	2.00
1:3.5	19.778	15.876	3.14	2.67
1:4	28.756	22.321	3.82	2.97

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. Due to the higher visual quality the system find applications in various real time applications such as Television processing, video conferencing, Internet video processing, Tele medicine etc.

7. REFERENCES

[1] S. E. Reichenbach and F. Geng, "Two- dimensional cubic convolution," IEEE Trans. Image Processing, vol. 12, No.8, pp. 857-865, August 2003.

[2] M. Unser, A. Aldroubi, and M. Eden, "B-spline signal processing: Part II - Efficient design and applications," *IEEE Transactions on Signal Processing*, vol. 41, no. 2, pp. 834-848, Feb. 2003.

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

[4] 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.

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

[6] 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.

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

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

[9] 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.

[10] 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.

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

[12] 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.

[13] H. S. Hou and H. C. Andrews,"Cubic spline for image interpolation and digital filtering," *IEEE Trans. Acoustics Speech Signal Processing*, vol. ASSP-26, pp. 508–517, Dec. 1978.

[14] M. Unser, A. Aldroubi, and M. Eden, "B-spline signal processing: Part I - Theory," *IEEE Transactions on Signal Processing*, vol. 41, no. 2, pp. 821-833, Feb. 1993.

[15] L. J. Wang, W. S. Hsieh, T. K. Truong, "A fast computation of 2D cubic-spline interpolation," *IEEE Signal Processing Letters*, vol.11, no.9, pp.768-771, Sept. 2004.