

A robust combination interpolation method for video super-resolution

- **Bui Thu Cao**

Ho Chi Minh City University of Industry (HUI)

- **Le Tien Thuong**

- **Do Hong Tuan**

University of Technology, VNU-HCM

- **Nguyen Duc Hoang**

Broadcast Research and Application Center, Vietnam Television (VTV-BRAC)

(Manuscript Received on July 10th, 2012, Manuscript Revised June 05th, 2013)

ABSTRACT:

This paper presents an efficient method for video super-resolution (SR) based on two main ideals: Firstly, input video frames can be separated into two components, non-texturing image and texturing image. Then each component image is applied to a compatible interpolation method to improve the quality of high-resolution (HR) reconstructed frame. Secondly, based on the approach that border regions of image details are the most lossy information regions from the sampling process. Therefore, a task of compensation interpolation is essential to increase the quality of the reconstructed HR

images. From these discussions, we proposed an efficient method for video SR by combining the spatial interpolation in different texturing regions and the sampling compensation interpolation to improve the quality of video super-resolution. Our results shown that, the quality of HR frames, reconstructed by the proposed method, is better than that of other methods, and in recently. The significant contribution is the low complexity of the proposed method. Hence, it is possible to apply the proposed algorithm to real-time video super-resolution applications.

Keywords: Video Super-Resolution, Image Super-Resolution.

INTRODUCTION

Video super-resolution is to reconstruct and create HR video frames from the input low-resolution (LR) video frames. According to the purpose of increasing in quality of image information, video SR is recently interested as an important research direction. Up to now, there are many authors with their methods for image SR reconstructions, as described in technical

overview of Park in 2003. In general, there are two types of SR methods, single-frame SR and multi-frame SR.

In single-frame SR, these methods use interpolation techniques in spatial or frequency domain to upscale the input LR frame. Then the reconstructed HR image is applied by filtering, smoothing and reshaping techniques to decrease

noises and increase quality of the reconstructed HR image. There are some typical studies. Li in 2001 used New Edge-Directed Interpolation (NEDI) to interpolate HR images in the wavelet domain. Takeda in 2007 developed a frame work for SR image using Multi-Dimension Kernel Regression Interpolation (KRI). In this method, each pixel in the video frame sequence is approximated with a 2-D local Taylor series. Mallat in 2010 used Sparse Mixing Estimators (SME) to define coefficients for interpolating in the wavelet transform. W. Dong [4] in 2011 used adaptive sparse domain selection and adaptive regularization (ASDS) to interpolate HR images in spatial domain.

In multi-frame SR, the input frames are registered the motion between them. Then based on the registered parameters, the input frames are rearranged in the same co-ordinate. The image information missed in the sampling process will be combined to recover the HR original image. There are some typical studies in multi-frame SR. Keren in 1988 based on the first order Taylor expansion to solve the registration equations. Vandewalle in 2006 and Bui-Thu in 2009 are based on the fact that two shifted images, which are different in the frequency domain only by a phase shift, can be found the shifts from their correlation in the Fourier transform. Lui , in

2011, also has achieved significant progress results. The author has proposed a Bayesian approach for adaptive video super-resolution. The proposed algorithm estimates simultaneously the motion of the details, noise kernel and noise level, while reconstructing the HR frame.

There are many input data for solving the reconstruction problems. Therefore, the multi-frame methods are usually more efficient than the single frame methods, and they are possible to reconstruct HR frames in higher quality. However, multi-frame SR methods take more time than single-frame methods for processing time. Thus, it is impossible to apply multi-frame SR for video applications, which demands real-time processing.

Although there are many researches in single-frame SR with advanced results, but they still exist two key problems. Firstly, single-frame SR methods usually create degradation at the ledge of texturing details, as what we see in Figure 1. Therefore, the advanced algorithms have to solve enough good for this problem. Secondly, most recent advanced algorithms can provide high quality in reconstructed HR frame. However, they also take too much time for SR process, so it is impractical for applying the recent SR algorithms to real-time SR video processing.



Ảnh được nội suy Bicubic
PSNR = 32.4dB and SSIM = 0.954



Ảnh được khôi phục bằng KRI
PSNR = 30.5dB and SSIM = 0.939



Ảnh được khôi phục bằng SME
PSNR = 32.0 dB and SSIM = 0.962



Ảnh được khôi phục bằng ASDS
PSNR = 32.0 dB and SSIM = 0.962

Figure 1. Illustrates the degradation at ledges of texturing details of the reconstructed HR image by typical single-frame SR algorithms

Our Approach

With the aim of our study for video super-resolution, we have to solve the two key problems by decreasing the degradation at the ledge of texturing details and directing to real-time processing for the proposed SR algorithm.

In order to decrease the degradation and increase the quality of the reconstructed HR frame, this paper presents an efficient method for single-frame video SR based on two main ideas: Firstly, input video frames are separated into two components, low-frequency image and high frequency image. Then, compatible interpolation method is applied to each component to improve the quality of HR reconstructed image. Secondly, border regions of image details are the most lossy information regions from the sampling process. Therefore, a task of compensation interpolation is essential to increase the quality of reconstructed HR image. Based on these ideas, we proposed an efficient method for video SR by combining the spatial interpolation in different frequency domains and the sampling compensation

interpolation for improving the quality of SR video images.

To directing to real-time processing, the proposed algorithm is innovated from the Cubic interpolation technique.

Overview of Paper

The structure of the paper organizes as follow: in section II, we propose the Spatial Interpolation in Different Texturing Regions method (SIDTR). Next, to increase the accuracy, in section III, we propose the Sampling Compensation Interpolation method (SCI) for the HR image, reconstructed in the section II. In the section IV, we propose the Combining Spatial Interpolation methods (CSI) by combining SIDTR and SCI. The results are present by comparing to different algorithms. In section V, we release the conclusion.

2. RELATION WORKS

2.1. Bilinear interpolation

In mathematics, Bilinear interpolation [10] is an extension of linear interpolation for a uniform 2-dimensional image space.

It is supposed that we want to explain the value of a function f at unknown points $P(x, y)$, but we know 4 points which belong the function: $f, Q_{11} = (x_1, y_1), Q_{12} = (x_1, y_2), Q_{21} = (x_2, y_1)$ và $Q_{22} = (x_2, y_2)$.

The method is linear interpolation in one direction, then further interpolated for the remaining directions, and the f function is developed as follow:

$$f(P) \approx \frac{y_2 - y}{y_2 - y_1} \left\{ \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}) \right\} + \frac{y - y_1}{y_2 - y_1} \left\{ \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}) \right\} \quad (1)$$

2.2. Cubic interpolation

To overcome the shortcomings of linear interpolation, cubic interpolation [10] is developed. It is based on the concept that the relationship between gray level values of pixels is nonlinear, and expressed as a polynomial of type 3, as follows:

$$f(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 (a_{ij} x^i y^j) \quad (2)$$

One of the characteristics of the cubic polynomial interpolation, we need 16 points surrounding a point (x, y) to solve out the parameters a_{ij} . Based on these parameters, we determine $f(x, y)$. Solving for the images on the large size requires a lot operation, as well as time consuming for processing.

In practical, to increase speed of the algorithm also as grow quality of the interpolated image, the cubic algorithms were developed follow as piecewise cubic polynomials. The piecewise functions, with format as follows:

$$P(s) = \begin{cases} P_1(s) & \text{if } s_1 \leq s \leq s_2 \\ P_2(s) & \text{if } s_2 \leq s \leq s_3 \\ \dots & \dots \\ P_{n-1}(s) & \text{if } s_k \leq s \leq s_{k+1} \end{cases} \quad (3)$$

With $s = s(x, y)$ is co-ordinate of the pixels,

P_i is piecewise polynomials, which are describes as follow:

$$P_i(s) = a_i(s - s_i)^3 + b_i(s - s_i)^2 + c_i(s - s_i) + d_i$$

$$i = 1, 2, \dots, n - 1 \quad (4)$$

Based on their ability of derivative and continuous boundary conditions between the adjacent pixels we can solve and determine the parameters of $P(x)$.

There are two efficient cubic interpolation algorithms which have been developed in Matlab [10]. They are Bicubic Interpolation and Cubic Spline Interpolation.

2.2.1. Bicubic interpolation

Bicubic interpolation in Matlab use Piecewise Cubic Interpolation Hermic (Pchip). The algorithm is presented as follows:

Set h_k as the distance of the k^{th} subpixel,

$$h_k = s_{k+1} - s_k \quad (5)$$

Let δ_k the first order different of $P(s)$, we have:

$$\delta_k = \frac{y_{k+1} - y_k}{h_k} \quad (6)$$

Let d_k the slope of the interpolated function, $P(s)$. We get:

$$d_k = P'(s_k) \quad (7)$$

The piecewise cubic interpolation of $P(s)$ in space $s_k \leq s \leq s_{k+1}$ is:

$$P(s) = \frac{3hp^2 - 2p^3}{h^3} y_{k+1} + \frac{h^3 - 3hp^2 + 2p^3}{h^3} y_k + \frac{p^2(p-h)}{h^2} d_{k+1} + \frac{p(p-h)^2}{h^2} d_k \quad (8)$$

With $p = s - s_k$ and $h = h_k$ in range of $s_k \leq s \leq s_{k+1}$, and (3.16) has to satisfy four conditions, as follow:

$$P(s_k) = y_k, P(s_{k+1}) = y_{k+1}, P'(s_k) = d_k, P'(s_{k+1}) = d_{k+1} \quad (9)$$

From (8) and four conditions of (9), we can find out the parameters of d_{k+1} and d_k . Based on the parameters we can define the Pchip interpolation function of $P_i(s)$.

2.2.2. Cubic Spline interpolation

Cubic Spline interpolation was developed on the basis of interpolation Pchip. This method is added smoothing by using continuous conditions at the curving points. The contents of the following methods:

From (8), we have the second derivative of $P(s)$,

$$P''(s) = \frac{(6h-12p)s_k + (6p-2h)d_{k+1} + (6p-4h)d_k}{h^2} \quad (10)$$

At $s = s_k, p = 0$, we have the second derivative in negative direction of $P(s)$ is,

$$P''(s_k+) = \frac{6d_k + 2d_{k+1} - 4d_k}{h_k} \quad (11)$$

At $s = s_{k+1}, p = h_k$, we have the second derivative in positive direction of $P(s)$ at s_{k+1} ,

$$P''(s_{k+1}-) = \frac{-6d_k + 4d_{k+1} + 2d_k}{h_k} \quad (12)$$

Similarly, we have the second derivative in negative direction of $P(s)$ at s_k is,

$$P''(s_k-) = \frac{-6d_{k-1} + 4d_k + 2d_{k-1}}{h_{k-1}} \quad (13)$$

The continuous condition at the curving point or also called the curving condition, at s_k ,

$$P''(s_k+) = P''(s_k-) \quad (14)$$

From (8), (9) and (14), we can solve out the values of parameters: d_k, d_{k+1} .

2.3. Evaluations about the spatial interpolations

Bilinear interpolation is shown that the simplicity of its algorithm with the linear relationship between the gray levels of pixels. So when the image is interpolated, the detail regions which have the gray values varying linearly have results better than the detail regions which have the gray level values varying non-linearly. Bicubic interpolation has been developed to overcome the defect of Bilinear interpolation. It is good mapping ability for image space. However, Bicubic interpolation is not enough good for smoothing image while Cubic Spline interpolation is stronger than Pchip interpolation for smoothing image by using the curving condition at each pixels. Both Bicubic and Cubic Spline interpolation have low complexity and fast processing time. These methods have been using in practical for real-time processing video applications.

It can be seen in Figure 2, which illustrates the response of the spatial interpolation techniques. In the area of detail which has gray level variable brokenly, the Cubic Spline interpolation reconstructs of the signal curve better than Bicubic (Pchip) interpolation. In other words, Cubic Spline interpolation allows restoring high frequency components from sampled images better than Bicubic interpolation. However, when applied to the texturing details of image, Cubic Spline interpolation will get the results less than Bicubic (Pchip) interpolation. We can see this illustration in Figure 3. In the border areas of texturing details, where there is mutation of the gray values, the Cubic Spline interpolation created degradation.

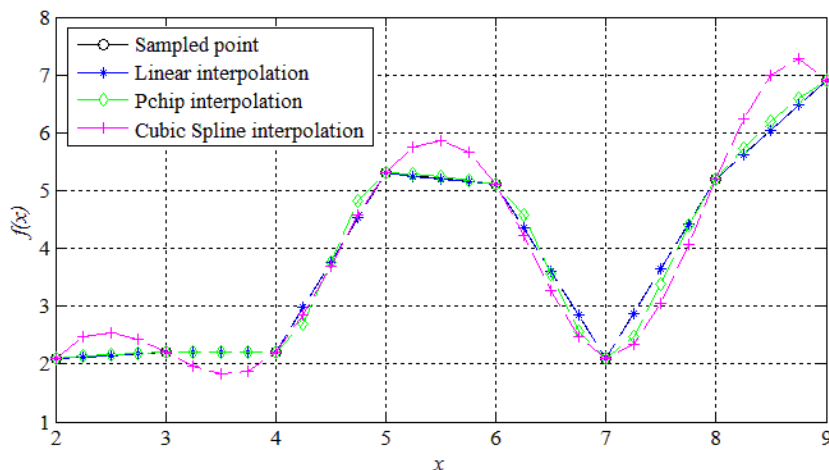


Figure 2. Illustrate the respond of spatial interpolation techniques

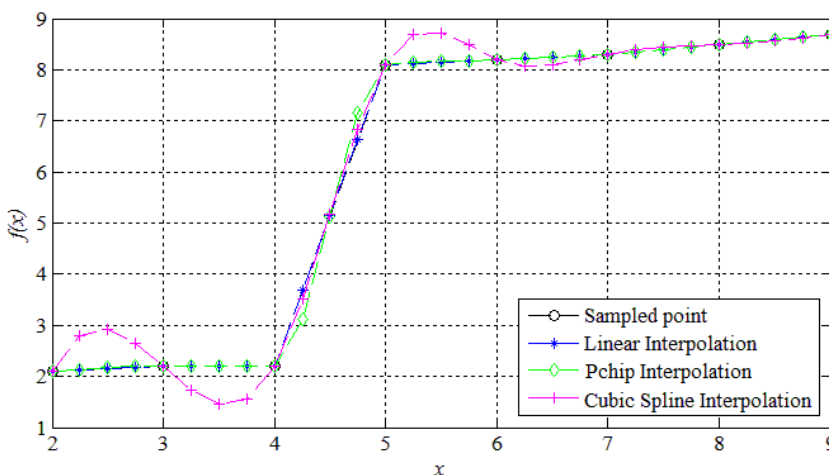


Figure 3. Illustration errors of different spatail interpolation methods at border regions

3. SPATIAL INTERPOLATION IN DIFFERENT TEXTURING REGIONS

Spatial interpolations are very useful in SR image reconstruction for increasing the quality as well as decreasing the processing time. To simulate this advantage, the standard video sequences are down-sampled in scale 2x2, to create LR video sequences. Then the LR video sequences is interpolated to upscale by different algorithms, with scale 2x2, to create HR frames. The PSNR measurement is used to evaluate the

quality of different algorithms. As seen in Table I, the quality of the advanced algorithms is not much higher than that of Bicubic algorithm. However, the processing time of Bicubic algorithm is very fast to compare with that of the others. The average processing time for up-scaling 30 frame sequences in size 144x176 pixels, by CPU Core 3i 2.53 GHz is 600 seconds for NEDI [2] , 1 seconds for Bicubic, 200 seconds for KRI [3], and 1200 seconds for SME [4].

Base on the above evaluations, we proposed a robust spatial interpolation algorithm by spatial interpolation in different texturing regions (SIDTR). The proposed algorithm uses low-pass filter to separate the texturing detail image from the image frame. We get texturing image and

non-texturing image. Next it is used the Linear interpolation for texturing image, and the Cubic Spline interpolation for none-texturing image. Then, combining two interpolated images, we get a HR reconstructed image.

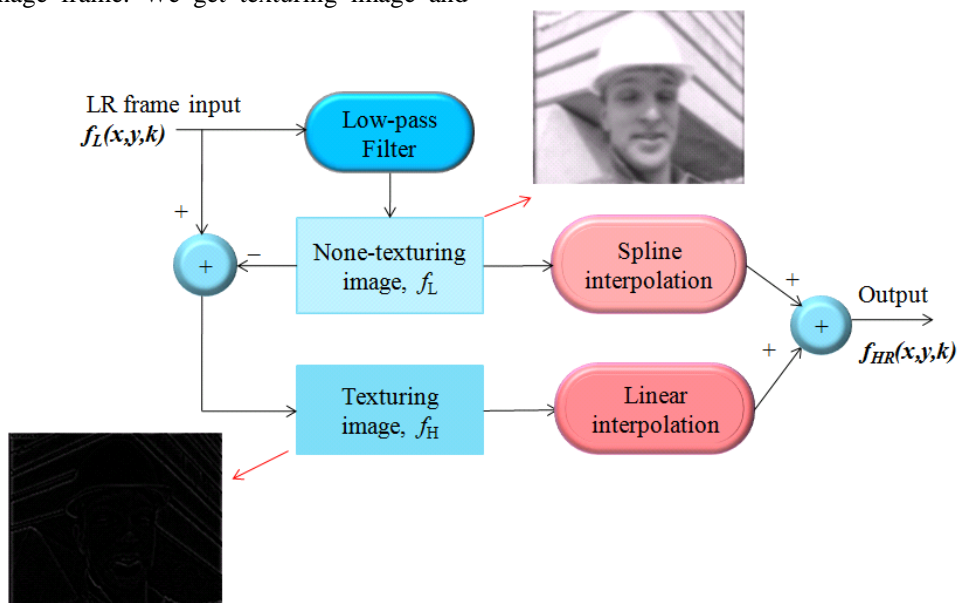


Figure 4. Illustration of the spatial interpolation in different frequency domains method

The proposed method is implemented in the block diagram at Figure 4. For each mono color space of the frame input, firstly, the input video frames are filtered by Low-pass Gaussian Filter. The output image of Low-pass Filter is non-texturing image, f_L . Then subtracting the original frame with the non-texturing image, f_L , we get the texturing image f_H . Next, the texturing image is interpolated by using linear interpolation method, and the non-texturing

image is interpolated by using Cubic Spline method, with scale of 2×2 . Finally, the two interpolated images are added to create a nature HR image.

To find the optimum cutoff frequency for the Low-pass Gaussian Filter we implemented the SIDTR method for nine standard video sequences. The results are shown in Figure 5. The optimum cutoff frequency is selected about 20 to 30.

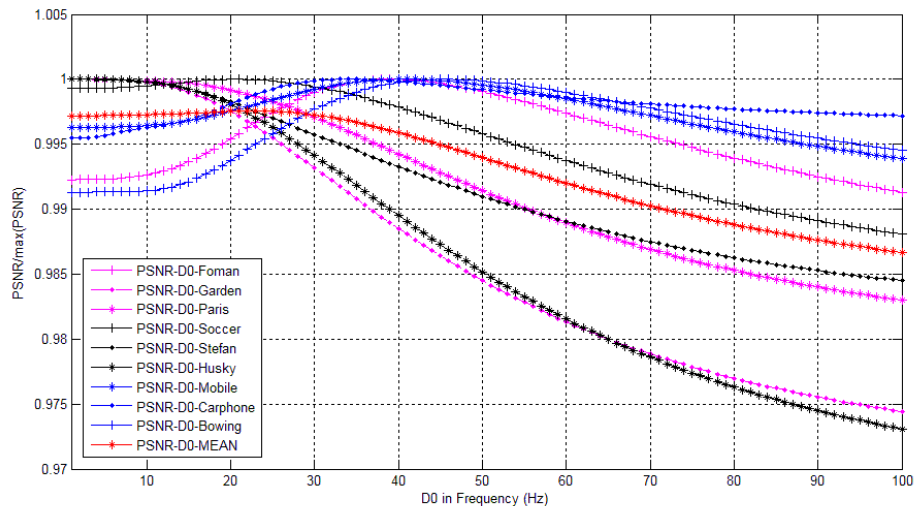


Figure 5. Statistic the gain of PSNR versus the frequency cutoff D_0

4. SAMPLING COMPENSATION INTERPOLATION

When sampled, images usually lose much the detail information at border pixels. As illustrated in Figure 6.a), for the sampled image, the sampled positions of pixels are in red points. Figure 6.b) show the image after being sampled. The sub-pixels in green points are lossy information of border regions. It is easy to realize that if the sampled image is zoomed in then the visual quality at the sub-pixels of border regions will be degraded. Consequently, to increase the quality for the upscaled image, we have to interpolate compensation for sampling process.

Through the experimental statistics, we proposed four types of sampling compensation

interpolation. For the type I, as shown in Figure 7 a) & b), the above border pixels are in light blue and the below border pixels are in dark blue. The gray levels of pixels, which are called in the same border, are approximate to each other and far different from the gray levels of the opposite pixels. Position 1 and 2 are base-points to interpolate. The condition of border pixels, at the base-point position 1, $P(x, y)$, is present as follow,

$$\begin{cases} |f_g(x, y+1) - f_g(x, y)| < Threshold1 \\ |f_g(x-1, y) - f_g(x, y)| < Threshold1 \\ |f_g(x-2, y) - f_g(x, y)| > Threshold2 \\ |f_g(x-1, y+1) - f_g(x, y)| > Threshold2 \end{cases} \quad (15)$$

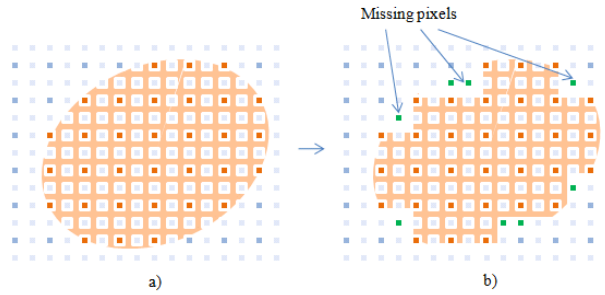


Figure 6. a) Sampled pixels at the red points, b) the loss information at the green points.

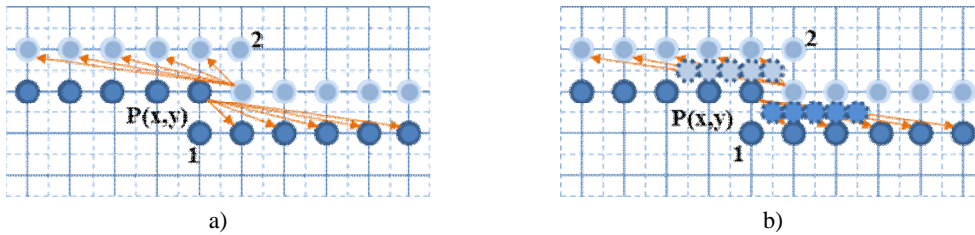


Figure 7. Interpolation directions of type I, at two based-points 1 and 2, in orange vectors

The thresholds are defined based on the standard mean and deviation of gray level differentiation, as following:

$$\begin{aligned} \text{threshold1} &= \mu + \sigma, \text{ and} \\ \text{threshold2} &= \text{threshold1} + C \end{aligned}$$

With,

$$\mu = \frac{1}{(M-1) \times (N-1)} \left[\sum_{x=1}^{M-1} \sum_{y=1}^{N-1} (f(x,y) - f(x+1,y+1)) \right]$$

$$\sigma = \sqrt{\frac{1}{(M-1) \times (N-1)} \left[\sum_{x=1}^{M-1} \sum_{y=1}^{N-1} (f(x,y) - \mu)^2 \right]}$$

C is a threshold to discriminate the border region between details of the image. Refer to intra prediction algorithms for video compression. We selected C by 10 (for the range of gray levels from 0-255).

Figure 7.a) illustrates the interpolation directions of type I, at two base-point 1 and 2, in orange vectors. At the base-point 1, in region of below border pixels, have interpolation directions: 45°, 26.5°, 18.4°, 14°, and 11.3°. At the base-point 2, in region of above border pixels,

have interpolation directions: 225°, 206.5°, 198.4°, 194°, and 191.3°. Figure 7.b) illustrates the positions of the interpolated subpixels, as middle blue points.

Figure 8 presents type II of border, with the base-points to interpolate at the position 3 and 4. Figure 9 shows type III and IV of border, with the base-points at the positions 5, 6, 7 and 8. Similarly, It is easy for us to find out the border conditions and border interpolation algorithms of the other base-point pixels.

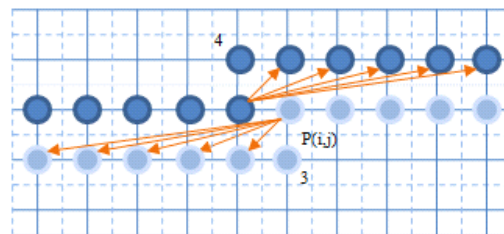


Figure 8. Illustrates the interpolation directions of type II in orange vectors. At the base-point 3, in region of the below border pixels, have the interpolation directions: 135°, 153.5°, 161.6°, 166°, and 168.7°. At the base-point 4, in region of above border pixels, have

the interpolation directions: -45° , -26.5° , -18.4° , -14° and -11.3° .

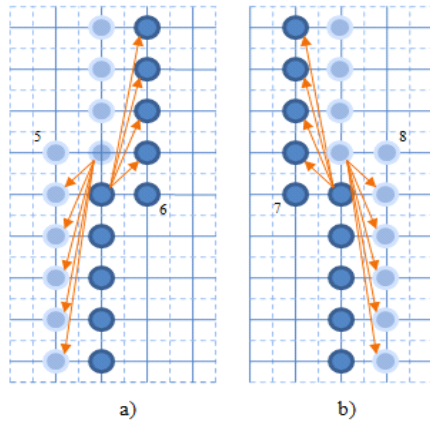


Figure 9. a) The interpolation directions of type III, b) the interpolation directions of type IV

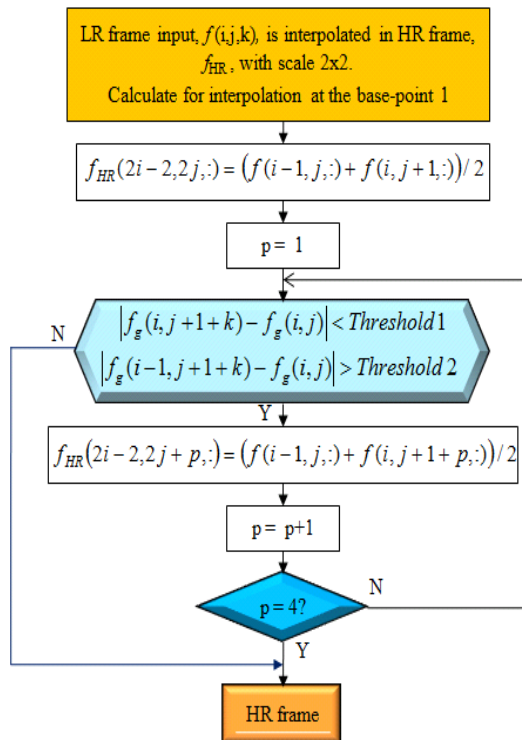


Figure 10. Illustrates the sampling compensation interpolation algorithm for the border type I, at the base-point 1, for the sub-pixels at the below border.

Figure 10 presents the sampling compensation interpolation algorithm for the sub-pixels in region of the below border pixels, at the position 1 of Figure 7. The input LR frame, $f(x,y,k)$, in dark-blue grid, is interpolated into HR frame, f_{HR} , in light-blue grid. The k is present for the processing mono color space (R,G,B or Y,U,V). The sub-pixels are interpolated in directions, which are in orange vectors, corresponding to parameters, p . The maximum value of p is 4, which is selected from practice about discriminating ability of the eye for straight edge.

5. RESULTS OF WORK

To evaluate the result of the proposed interpolation method, we implemented practical experiments on eight standard sequences, as shown in Figure 10. To present power of the proposed algorithm, the video standard sequences were selected in form variety of real image details, from less detail sequences as: Foreman, Soccer and Pamphlet, to more detail sequences as: Mobile, Paris, Stefan, Flower-garden and Husky. The more details video sequence has in, the more complexity program has to solve.

Firstly, the video frame sequences are down-sampled, with scale of 2×2 , to create input LR frames. Then the LR frames are up-scaled, with scale of 2×2 , by the proposed method, SIDTR, as present in section 3, to create HR frame. Next, the reconstructed HR frames are interpolated for sampling compensation by using SCI method to increase the quality of the final reconstructed HR frames, as presented in section 4. To evaluate the quality of HR reconstructed frame, we use PSNR and SSIM [11] measurement between the original HR frames and the HR frames reconstructed by different algorithms.

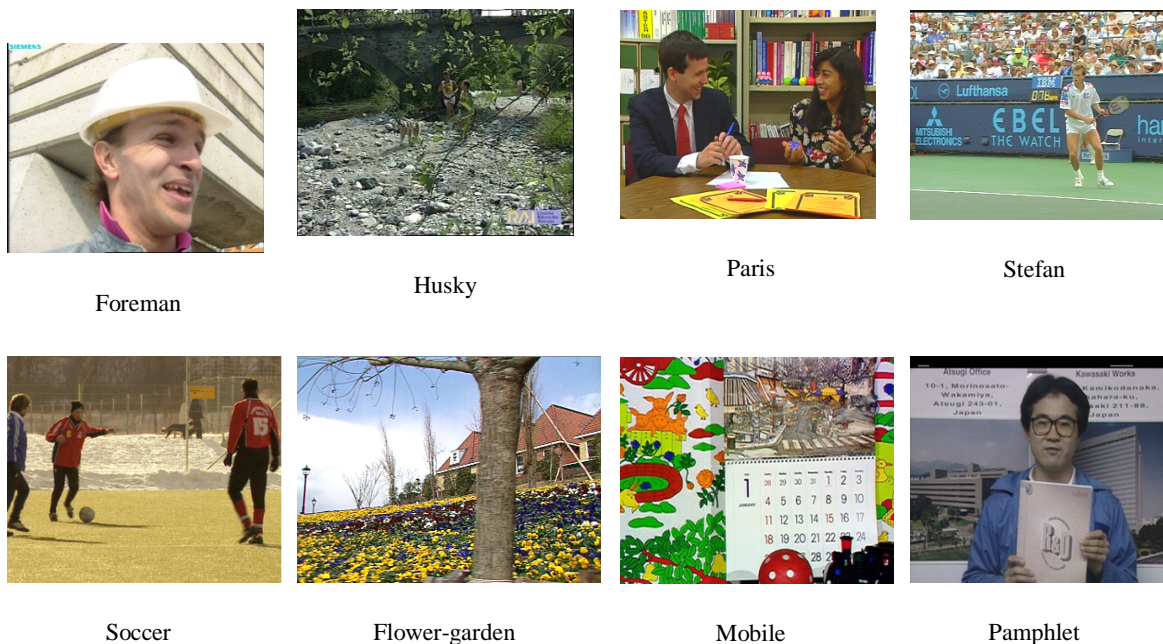


Figure 10. Standard video sequences are used for the practical experiments.

PSNR Results of the proposed methods are present in two columns, SIDTR and CSI, as shown in Table 1. The SIDTR column is the results of the proposed method in section 3. The CSI column is the results of the final proposed method, combining of spatial interpolation methods, SIDTR and SCI. The results of the proposed method, SIDTR, are better to compare that of other algorithms. Furthermore, the SCI method is proved even more efficient by higher the PSNR than that of SIDTR. The average PSNR values of the proposed method CSI are

higher than that of other algorithms, 0.27 dB to Bicubic method, 1.57 dB to KRI' results, 0.09 dB to SME' results, and 0.14 dB to ASDS' results. Although our results are slightly higher than that of the SME algorithm, but our processing time is much faster than that of SME algorithm. The average processing time for one frame, in size 144x176 pixels, is about **0.36 seconds** for the proposed algorithm (CSI), **0.3 seconds** for Bicubic, while it takes about **243 seconds** with the SME algorithm, and **39 seconds** with the ASDS algorithm.

Table 1. Comparison of mean PSNR (db) of different algorithms, Bicubic (Pchip), KRI, SME, ASDS, and the proposed methods

| Author\ Sequence(frame) | Bicubic | KRI | SME | ASDS | SIDSR | CSI |
|----------------------------|---------|-------|--------------|--------------|-------|--------------|
| Foreman(1-10) | 31.49 | 29.92 | 32.63 | 31.10 | 31.63 | 32.51 |
| Husky (1-10) | 17.09 | 16.73 | 17.14 | 18.03 | 17.34 | 17.38 |
| paris (1-10) | 21.59 | 20.74 | 21.66 | 22.68 | 21.71 | 21.81 |
| Stefan (1-10) | 22.44 | 21.65 | 22.31 | 23.54 | 22.66 | 22.61 |

| | | | | | | |
|-----------------------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Soccer (1-10) | 28.09 | 26.77 | 28.18 | 28.33 | 28.24 | 28.26 |
| Flower-Garden (1-10) | 18.70 | 18.25 | 18.91 | 19.66 | 18.94 | 18.98 |
| Mobile(1-10) | 21.38 | 19.77 | 21.42 | 21.35 | 21.40 | 21.35 |
| Pamphlet(1-10) | 31.61 | 28.14 | 31.57 | 30.80 | 31.51 | 31.65 |
| Average gain | -0.27 | -1.57 | -0.09 | 0.11 | -0.14 | 0 |

Table 2. Comparison of mean SSIM (in db) of different algorithms, Bicubic (Pchip), KRI, SME, ASDS and the proposed method (CSI)

| Author\ Sequence(frame) | Bicubic | KRI | SME | ASDS | CSI |
|-----------------------------|---------------|---------------|---------------|---------------|--------------|
| Foreman(1-10) | 0.952 | 0.937 | 0.963 | 0.951 | 0.962 |
| Husky (1-10) | 0.672 | 0.554 | 0.676 | 0.671 | 0.679 |
| paris (1-10) | 0.853 | 0.794 | 0.863 | 0.855 | 0.862 |
| Stefan (1-10) | 0.850 | 0.784 | 0.847 | 0.855 | 0.851 |
| Soccer (1-10) | 0.868 | 0.809 | 0.866 | 0.858 | 0.873 |
| Flower-Garden (1-10) | 0.763 | 0.678 | 0.774 | 0.767 | 0.768 |
| Mobile (1-10) | 0.837 | 0.726 | 0.836 | 0.805 | 0.834 |
| Pamphlet(1-10) | 0.958 | 0.897 | 0.959 | 0.946 | 0.958 |
| Average gain | -0.004 | -0.076 | -0.000 | -0.009 | 0 |

SSIM Results presents the structure similarity measurement [11] between the HR original frame and the HR frames reconstructed by different algorithms, as shown in Table 2. It shows that the proposed algorithm has the results better than that of order algorithms.

Moreover, we evaluate intuitively the quality of the HR frames reconstructed by the proposed method and the best algorithm, SME algorithm. It can be seen that at border regions, the HR image frames reconstructed by the proposed method CSI is more clearly in detail than that of the SME algorithm.

6. CONCLUSION

The key problem of any super-resolution algorithms is the degradation at edge regions,

where have changes suddenly in gray level of pixels. And the proposed algorithm has solved that problem more efficiently by combining the spatial interpolation in different frequency ranges and the sampling compensation interpolation. It is seen that the results of the proposed algorithm are better than that of the recently algorithms, as of [3], [4] and [5].

The significant contribution of the proposed method is presenting a simple algorithm with less processing time. Therefore, it is absolutely to apply the CSI algorithm for real-time video super-resolution systems by integrating parallel processing in FPGA chips and multi-core processors.



Original HR Foreman image



Apart of the original HR Foreman



Reconstructed by Bicubic, 31.38dB



Apart of the Bicubic image



Reconstructed by SME, 32.56 dB



Apart of the SME image



Reconstructed by CSI, 32.58 dB



Apart of the CSI image

Figure 11. Comparison of SR results, applied on Foreman sequency at frame 11. From the top to botton, they are the original frame, the LR frame, the HR frame reconstructed by SME algorithm and the HR frame reconstructed by the proposed method, CSI.



HR Original Soccer frame



Apart of the HR original Soccer frame



Reconstructed by Bicubic, 28.24dB



Apart of the Cubic frame



Reconstructed by SME, 28.32 dB



Apart of the SME frame



Reconstructed by CSI, 28.35 dB



Apart of the CSI frame

Figure 12. Comparison of SR results, applied on Soccer sequence at frame 1. From the top to bottom, they are the original frame, the LR frame, the HR frame reconstructed by SME algorithm and the HR frame reconstructed by the proposed method, CSI.

Siêu phân giải video bằng phương pháp nội suy kết hợp

- **Bùi Thư Cao**

Trường Đại học Công Nghiệp Thành phố Hồ Chí Minh

- **Lê Tiến Thường**

- **Đỗ Hồng Tuấn**

Trường Đại học Bách khoa, ĐHQG-HCM

- **Nguyễn Đức Hoàng**

Trung tâm Nghiên cứu Ứng dụng Khoa học Kỹ thuật Truyền hình

TÓM TẮT:

Bài báo trình bày một phương pháp hiệu quả cho siêu phân giải video dựa trên hai ý tưởng chính. Thứ nhất, frame video đầu vào có thể được tách ra thành hai thành phần, ảnh có kết cấu và ảnh không kết cấu. Sau đó, với mỗi thành phần ảnh ta dùng phương pháp nội suy tương thích để nâng cao chất lượng của hình ảnh HR được khôi phục. Thứ hai, dựa trên quan niệm rằng, vùng đường biên của các chi tiết ảnh là những khu vực bị mất thông tin nhiều nhất từ quá trình lấy mẫu. Do đó, nội suy bồi hoàn lấy mẫu là điều cần thiết để tăng chất lượng của hình ảnh

HR được khôi phục. Từ các ý tưởng ở trên, chúng tôi đề xuất một phương pháp hiệu quả cho siêu phân giải ảnh video bằng việc kết hợp nội suy không gian trong các vùng kết cấu khác nhau với nội suy bồi hoàn lấy mẫu. Kết quả của chúng tôi chỉ ra rằng, chất lượng của ảnh HR được khôi phục bởi phương pháp đề nghị là tốt hơn so với các phương pháp tiến bộ gần đây, như , và . Điểm quan trọng là sự đơn giản trong thuật giải của phương pháp được đề xuất. Nhờ đó, có thể áp dụng giải thuật đề nghị cho các ứng dụng siêu phân giải video thời gian thực.

REFERENCES

- [1]. H. Takeda, S. Farsiu, and P. Milanfar, *Kernel Regression for Image Processing and Reconstruction*, IEEE Transactions on Image Processing, 16(2):349 - 366, (2007).
- [2]. S. Mallat and G. Yu, *Super-Resolution with Sparse Mixing Estimators*, IEEE Transactions on Image Processing, 19(11): 2889 - 2900, (2010).
- [3]. W. Dong, L. Zhang, G. Shi, and X. Wu, *Image Deblurring and Super-Resolution by Adaptive Sparse Domain Selection and Adaptive Regularization*, IEEE Transactions on Image Processing, 20(7):1838 - 1857, (2011).
- [4]. S.C. Park, M.K. Park, and M.G. Kang, *Super-Resolution Image Reconstruction: A technical Overview*, IEEE Signal Processing Magazine, 20:21-26, (2003).
- [5]. X. Li and M.T. Orchard, *New Edge-directed interpolation*, IEEE Transactions on Image Processing, 10(10):1521-1527, (2001).

- [6]. D. Keren, S. Peleg, and R. Brada, *Image sequence enhancement using sub-pixel displacement*, in *IEEE Conference in Computer Vision and Pattern Recognition*, pp. 742-746, (1988).
- [7]. P. Vandewalle, S. Susstrunk, and M. Vetterli, *A Frequency Domain Approach to Registration of Aliased Images with Application to Super-Resolution*, *EURASIP Journal on Applied Signal Processing*, pp. 1-14, (2006)
- [8]. C. Bui-Thu, T. Le-Tien, T. Do-Hong, and H. Nguyen-Duc, *An Efficiently Phase-Shift Frequency Domain Method for Super-Resolution Image Processing*, in *Proceeding of The 2009 International Conference on Advanced Technologies for Communications (ATC2009) IEEE-REV*, pp. 68-73, (2009).
- [9]. C. Lui and D. Sun, *A Bayesian Approach to Adaptive Video Super Resolution*, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 209 – 216, (2011).
- [10]. Cleve Moler, *Numerical Computing with MATLAB*. USA: MathWorks, (2004).
- [11]. Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, *Image Quality Assessment: From Error Visibility to Structural Similarity*, *IEEE Transaction on Image Processing*, 13(4):600-612, (2004).
- [12]. C. C. Hsieh, Y. P. Huang, Y. Y. Chen, and C. S. Fuh, *Video Super-Resolution by Motion Compensation Iterative Back-Projection Approach*, *Journal of Information Science and Engineering*, 27(3):1107-1122, (2011).