

RESEARCH

Open Access

Video resampling algorithm for simultaneous deinterlacing and image upscaling with reduced jagged edge artifacts

Du Sic Yoo, Joonyoung Chang, Chul Hee Park and Moon Gi Kang*

Abstract

In this paper, we propose a video resampling method for simultaneous deinterlacing and image upscaling. The proposed method is composed of two steps: the initial image magnification step and the edge enhancement step. In order to convert an interlaced image into a display format image, a filtering strategy, which resizes images with arbitrary ratios and reduces the overall computational load, is performed region adaptively using local characteristics such as motion or motionless regions. After the initial step, the proposed jagged edge correction (JEC) method is applied to the initially upscaled images to correct the stair-like artifacts (jagged edges) which are caused by ignoring any edge information in diagonal edge regions during the linear filtering process. Moreover, this method can be very useful for various upscaling applications to improve edge quality since it can be used in combination with other common interpolation techniques, such as cubic spline techniques. Experimental results show that the proposed method substantially reduces the jagged edges of the converted images and provides steep and natural-looking edge transitions.

Keywords: Jagged edge artifacts; Ringing artifacts; Transient improvement; Windowed sinc; Deinterlacing; Interpolation; Upscaling

1 Introduction

Over the last several decades, image resolution has rapidly increased, and advanced display devices like high-definition television (HDTV) have been developed to keep pace with these rapid changes. Even today, image resolution capabilities are still increasing and some companies have already introduced next-generation broadcasting systems such as *Super Hi-Vision* and *Ultra High Definition* TV systems. However, not all commercial media content can be provided in high resolution. There is a large amount of low-resolution video content, and many consumers want to view this content in full screen mode with their high-resolution display devices. Therefore, deinterlacing (or interlaced-to-progressive conversion (IPC)) and image upscaling are required to convert incoming low-resolution interlaced images into high-resolution progressive ones.

The deinterlacing and image upscaling procedures can be used to solve the problems of format and spatial conversions, respectively. These procedures are called video-to-display format conversions (VDFC) [1] in this paper.

Deinterlacing and image upscaling have been studied for decades. These methods can be roughly classified into linear filtering interpolation (LFI) and edge directional interpolation (EDI). Among the deinterlacing methods, LFI approaches [2-7] can be categorized into spatial (intra-field), temporal (inter-field), and spatio-temporal methods, according to the field information. In order to obtain progressive images, missing pixels have to be reconstructed using linear filters according to the spatial correlations, the temporal correlations, and both the spatial and temporal correlations in interlaced video sequences. Particularly, some algorithms [5-7] discover missing pixel values by interpolating the pixels along motion trajectories, because temporal correlation is dependent on motion information. Among the image upscaling methods, LFI approaches [8-12] design a particular interpolation kernel, which can be applied to the

*Correspondence: mkang@yonsei.ac.kr
Department of Electrical and Electronic Engineering, Yonsei University, Seoul 120-749, South Korea

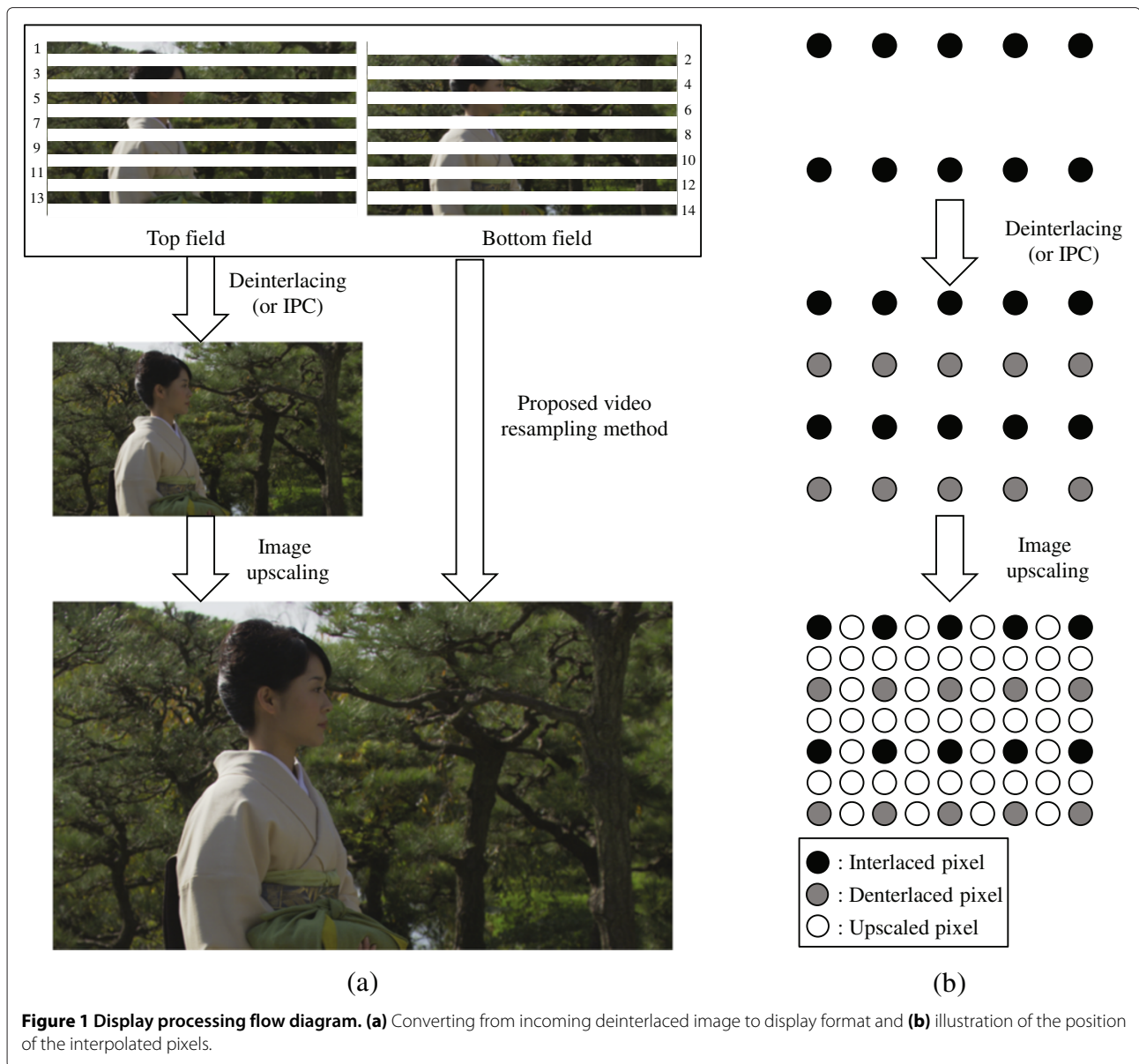
entire image. Especially, these methods can resize images with arbitrary ratios, which is one of the preferred features for image upscaling applications. LFI methods for both deinterlacing and upscaling are as old as image processing, and they are still popular because of their simple implementation. However, LFI approaches usually produce jagged edge artifacts (stair-like artifacts) in diagonal edge regions because they do not consider any edge information during the resampling process.

On the other hand, EDI techniques calculate the directional correlation between neighboring pixels to perform interpolation along estimated edge directions. These EDI approaches have been widely used for deinterlacing [13,14] and upscaling [15-17]. Especially, edge-dependent deinterlacing algorithms such as edge-based line average are the most popular among the intra-field deinterlacing methods. Many sophisticated techniques for both deinterlacing and upscaling have been proposed to increase the accuracy of the estimated edge directions because EDI approaches can improve visual quality around dominant edges when the estimated edge direction is correct. However, high computational loads are required to estimate the directions of the various diagonal edges, and thus, upscaling applications become more complicated than deinterlacing applications. Furthermore, for achieving arbitrary ratio enlargement, the EDI process generally becomes more complicated since edge direction estimation and interpolation are performed within an asymmetric interpolation lattice. Therefore, upscaling EDI methods usually restrict the scaling ratio in order to simplify the interpolation process, e.g., images are enlarged twice as much as the original images in both horizontal and vertical directions. With fixed scaling ratios, some researchers [15-17] estimate the covariance of high-resolution images by exploiting the covariance of low-resolution images. These methods can substantially improve the image quality by preserving the spatial coherence of the upscaled images. They also provide natural-looking images by efficiently connecting the disconnected edges. However, these algorithms may introduce some false edges or textures due to over-fitting. They can sometimes connect the edges erroneously due to their incorrect estimation of covariance because spatial correlations generally change after downsampling.

In consumer electronic devices such as HDTV systems, the VDFC technique is used to convert video sources into the kind of display resolution format shown in Figure 1. As shown in Figure 1b, deinterlacing doubles the vertical resolution of an interlaced image, while image upscaling resizes an image with arbitrary ratios. Therefore, image upscaling can be considered a generalized version of deinterlacing. Moreover, the VDFC technique is generally based on either LFI or EDI approaches. However, LFI approaches tend to suffer from jagged artifacts along the

diagonal edges despite offering the competitive advantages of low complexity and arbitrary ratio interpolation. As large display devices with high resolution become more popular, the visible artifacts become more prominent. Although EDI approaches produce good performance in edge regions, they require high complexity in order to offer spatial coherence of the upscaled images. Also, these methods may require additional LFI methods for achieving arbitrary ratio enlargement after performing the EDI process. Therefore, an efficient VDFC technique, which offers a low computation load and provides high edge quality, is required.

In this paper, we propose a video resampling method that takes the advantage of both the LFI and EDI methods: the capabilities of converting the image format, resizing the images with arbitrary ratios, and improving edge quality. The proposed method consists of two steps: initial image magnification and edge enhancement. In the first step, linear filtering strategies are performed differently to local region characteristics such as motion or motionless regions in order to convert interlaced images into upscaled progressive ones. The performance of the LFI is affected by the used interpolation kernels, and thus, an interpolation kernel closer to the ideal sinc function achieves better reconstruction performance. Thus, the conventional kernels are obtained by truncating the ideal sinc function. However, side lobes (or ripples) of the truncated kernel produce severe ringing artifacts within the interpolated images, and thus, the kernel size is limited to reducing the ringing artifacts. In order to cope with this problem, we use a ringing reduction technique to improve interpolation performance, which reduces the number of ringing artifacts while reconstructing the high-frequency components. Thus, the used linear filter is based on the Lanczos function, and the wider window function is applied to the Lanczos function in order to achieve better reconstruction performance. In the second step, the proposed jagged edge correction (JEC) method is performed to improve the edge quality of the initially upscaled images. More specifically, the JEC method uses an adaptive smoothing kernel obtained from the gradient covariance in order to correct the jagged edge artifacts within the initially upscaled images. This kernel, called an ellipsoidal kernel, assigns large weights along the local edge direction in order to smooth the jagged edge artifacts along the edge direction. In conventional EDI methods, the asymmetric interpolation lattice makes the interpolation procedure complex. However, the complexity level of the JEC method is less than that of the conventional EDI because the JEC procedure is applied to a symmetric lattice of the upscaled images. Moreover, enlarging the images resulted in blurring effects within the initially upscaled images, and these effects were perceived more conspicuously in regions with jagged edge artifacts. In



order to solve this problem, a transient improvement (TI) technique is performed during the filtering process along the edge direction to improve the sharpness of the initially upscaled images. The JEC method is a core technique of the proposed video resampling system, and it can be used as a postprocessor for many LFI methods, such as cubic spline. This independent module can be applied to various applications with flexibility.

The organization of the rest of this paper is as follows: In Section 2, previous works on Lanczos interpolation and related issues are discussed. In Section 3, the proposed video resampling method is described in detail. First, the overall structure of the proposed method is explained. In Section 3.1, an image magnification step based on the Lanczos function is presented, and the proposed ringing

artifact reduction technique is described to improve interpolation performance. Subsequently, the proposed JEC method is explained to improve the edge quality of the initially interpolated images. In Section 4, the experimental results of various test images are presented, and comparisons with other algorithms are made. Finally, conclusions are presented in Section 5.

2 Previous works on Lanczos interpolation

The normalized sinc function has been accepted as an ideal interpolation function that can perfectly pass low frequencies and perfectly cut high frequencies. However, this ideal interpolator has an infinite impulse response in the time and spatial domains so the sinc function has to be truncated or windowed in order to fulfill the

requirements of a hardware implementable interpolator. However, the truncated interpolation kernel produces severe ringing effects within interpolated images. Therefore, various windowing kernels such as Hann, Hamming, Cosine, Lanczos, Blackman, and Kaiser have been proposed to reduce these ringing effects. According to spectral analysis [12], these different windows display different spectral characteristics, and some tradeoffs occur when the window function is chosen. Thus, the choice of the windowing function is crucial, and it is very dependent on the selected applications. In Figure 2, we compare the sinc kernels truncated by various 6-tap windowing kernels. As shown in Figure 2a, each of the truncated sinc kernels has negative coefficients, which comes from the side lobes of a sinc function. These negative coefficients are used to produce steep and sharp edge transitions in the step edges and to recover the image details in the texture regions. Therefore, larger negative coefficients achieve better reconstruction performance in the edge regions. According to Figure 2b, the Lanczos windowing kernel has the most negative coefficients among the various windowing kernels. Thus, the Lanczos windowing kernel is preferred in order to improve the reconstruction performance of the high-frequency components.

The impulse response of the Lanczos interpolator is the normalized sinc function windowed by the Lanczos window, and the Lanczos window is the central lobe of a sinc function scaled to a certain extent. In one dimension, the Lanczos function can be obtained as

$$h_{Lz}(x, s) = \begin{cases} \frac{\sin(\pi x)}{\pi x} \frac{\sin(\pi x/s)}{\pi x/s}, & |x| < s \\ 1, & x = 0 \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

where $h_{Lz}(x, s)$ represents the Lanczos kernel and s is a positive integer (typically 2 or 3), which controls the

size of the Lanczos kernel. According to Figure 3, the Lanczos interpolator can achieve better reconstruction performance using wider windows (when s is set as 4, 8, or 12) since the interpolation kernels become similar to the ideal sinc function. However, if the window size increases, more side lobes of the ideal sinc function influence the interpolation kernels, and they cause unwanted artifacts such as ringing (or shooting) artifacts in flat regions, especially when high-frequency edges exist within the range of the wide window. From the above discussion, it is clear that the side lobes of an interpolation filter can improve reconstruction performance in edge regions, but they can also degrade the image quality because of the ringing artifacts.

3 Proposed video resampling method with jagged edge correction

Figure 4 roughly illustrates the overall structure of the proposed resampling method. As illustrated in Figure 4, the proposed method requires two steps to perform video resampling: the initial image magnification step and the edge enhancement step. In the first step, the image magnification process for deinterlacing and upscaling is performed one-dimensionally by performing one-dimensional interpolation processes in the horizontal and vertical directions separately. Interpolation kernels for directional interpolations are based on the Lanczos kernel with a ringing artifact reduction. Especially in the vertical interpolation process, temporal information is used and then the values of both modified spatio-temporal interpolation and modified Lanczos interpolation are mixed according to the motion detection process. In the second step, the proposed JEC method is applied to the initially upscaled images to improve the edge quality of the images. During the JEC process, a TI technique is also used to improve the sharpness of the upscaled images.

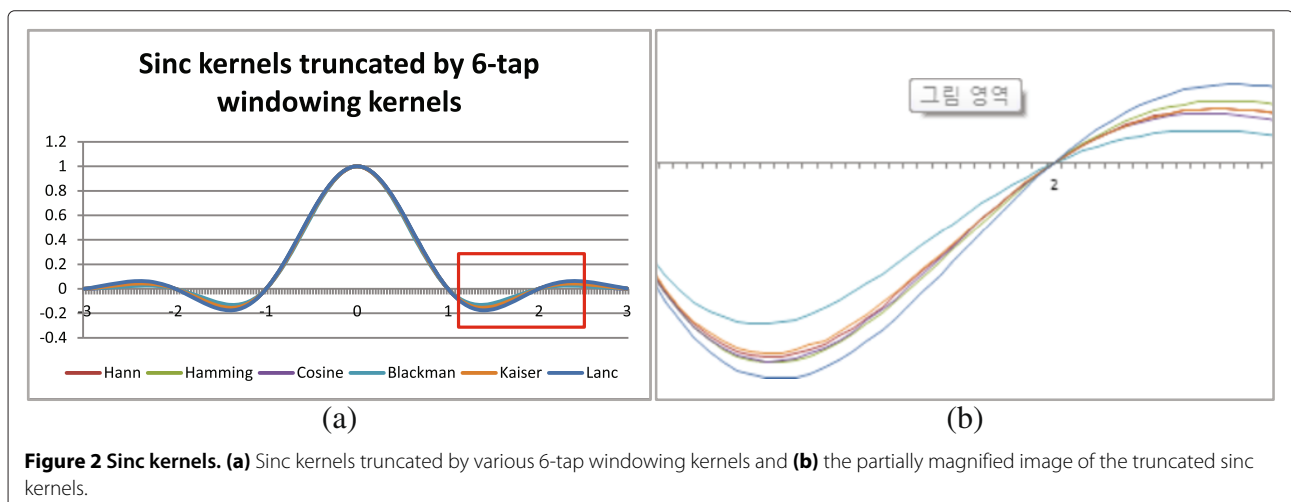


Figure 2 Sinc kernels. (a) Sinc kernels truncated by various 6-tap windowing kernels and (b) the partially magnified image of the truncated sinc kernels.

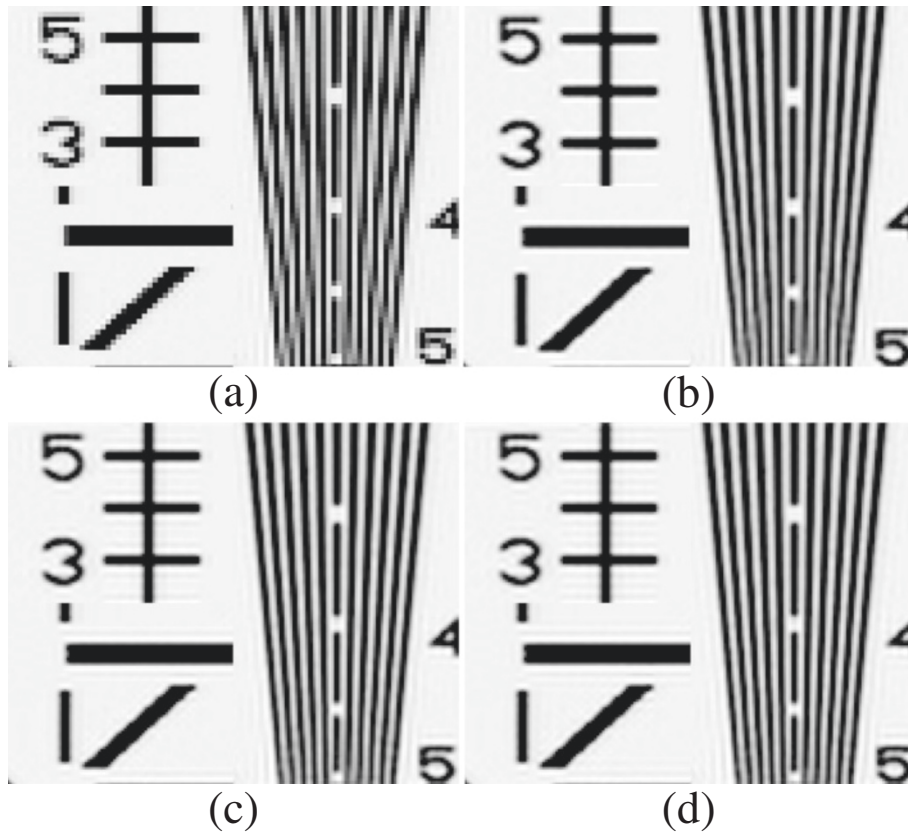


Figure 3 Upscaled images obtained by varying the size of the Lanczos kernel. (a) Input image. Filter as (b) $s = 4$, (c) $s = 8$, and (d) $s = 12$.

3.1 Initial image magnification step

3.1.1 Image magnification process with Lanczos function

The goal of the proposed resampling method is to convert an interlaced field to a scaled progressive frame. In order to achieve this goal, the image magnification process performs one-dimensional interpolation first in the horizontal direction and then again in the vertical direction. Based on the Lanczos function, the missing pixel value at an arbitrary position (Δ_i, Δ_j) is reconstructed by

$$\hat{f}_{Lz}^{if}(i+k, j+\Delta_j, n) = \sum_{l=-a+1}^a h_{Lz}(\Delta_j-l, a) \cdot f^{if}(i+k, j+l, n), \quad (2)$$

$$\hat{f}^{pf}(i+\Delta_i, j+\Delta_j, n) = \alpha \cdot \hat{f}_{Lz}^{pf}(i+\Delta_i, j+\Delta_j, n) + (1-\alpha) \cdot \hat{f}_{ST}^{pf}(i+\Delta_i, j+\Delta_j, n), \quad (3)$$

where the superscripts if and pf denote the interlaced image and the progressive image, respectively, and the subscripts Lz and ST represent the results of Lanczos interpolation and spatio-temporal interpolation, respectively. f^{if} and \hat{f}^{pf} represent the input interlaced image

and the output upscaled progressive image, respectively. \hat{f}_{Lz}^{if} denotes the horizontally upscaled interlaced image, and \hat{f}_{Lz}^{pf} and \hat{f}_{ST}^{pf} represent the upscaled progressive images obtained from Lanczos and spatio-temporal interpolations, respectively. (i, j) and n represent the spatial indices and the temporal index of the input interlaced image, respectively. a represents the horizontal size of the Lanczos kernel. Δ_i and Δ_j ($0 \leq \Delta_i, \Delta_j < 1$) represent the arbitrary positions to be interpolated in the vertical and horizontal directions, respectively. Since the pixels of the top and bottom fields are positioned alternately in the vertical direction, the relative vertical positions of the current, previous, and next fields are determined along the temporal index. Given the input interlaced image f^{if} , Δ_i and Δ_j are represented as

$$\Delta_i = \begin{cases} i \cdot sr_v - \lfloor i \cdot sr_v \rfloor - (n\%2)/2 & \text{if } n, (n \pm 2)\text{fields} \\ i \cdot sr_v - \lfloor i \cdot sr_v \rfloor & \text{if } (n \pm 1)\text{fields} \end{cases} \quad (4)$$

$$\Delta_j = j \cdot sr_h - \lfloor j \cdot sr_h \rfloor,$$

where sr_h and sr_v denote the horizontal and vertical scaling ratios, respectively, and $\lfloor \cdot \rfloor$ represents an operator to return the maximum integer less than or equal to the given

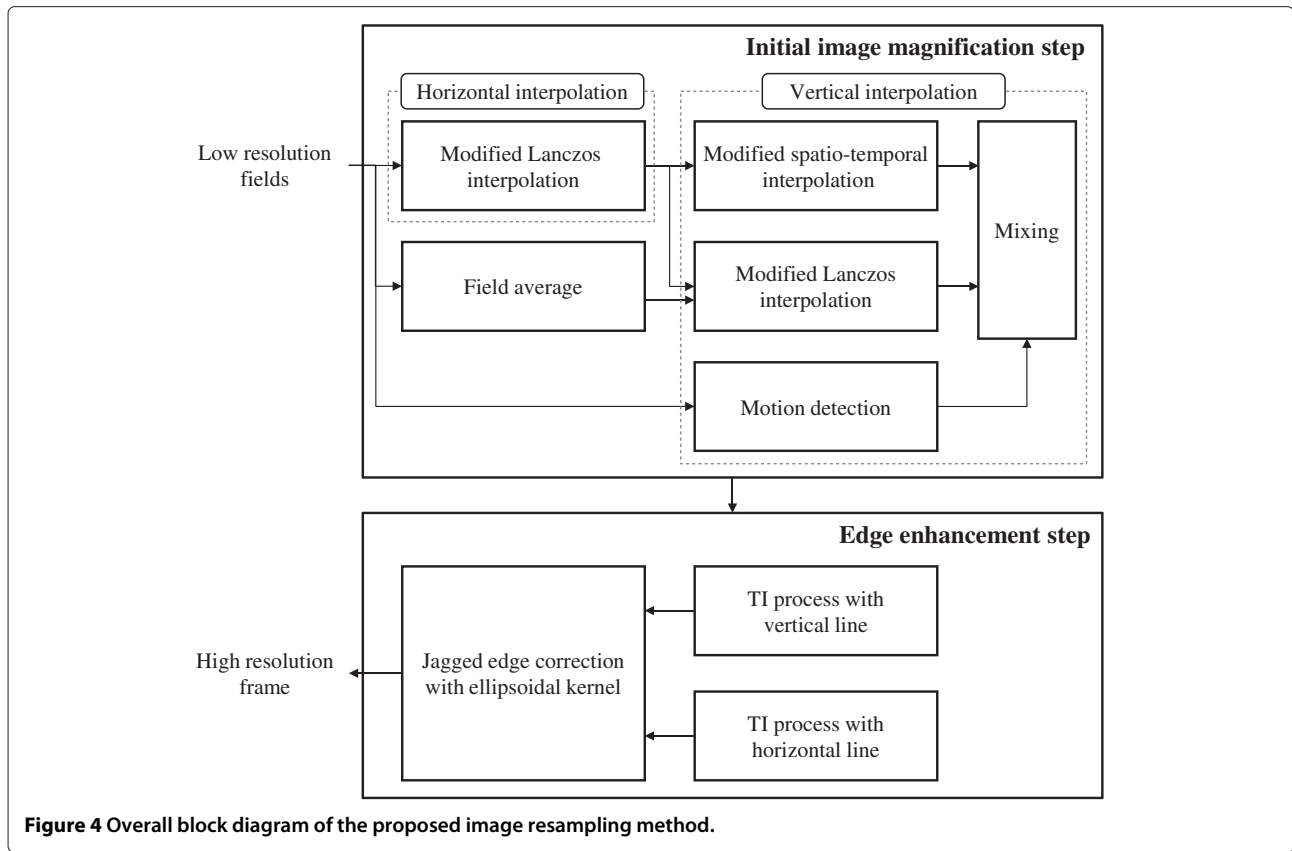


Figure 4 Overall block diagram of the proposed image resampling method.

real number. α of (3) is a weight to control the contribution of two values, \hat{f}_{Lz}^{pf} and \hat{f}_{ST}^{pf} . In the image magnification process, the horizontal scaling process is first performed using (2) and then the final results of the vertical interpolation are obtained from fusing Lanczos interpolation and spatio-temporal interpolation according to the motion detection process.

In the vertical interpolation process of (3), \hat{f}_{Lz}^{pf} and \hat{f}_{ST}^{pf} mainly contribute to \hat{f}^{pf} in the motionless and motion areas, respectively. These values are calculated using previous and next field information because the original pixel information at the missing position is obtained from the previous and next fields due to the inherent nature of the interlaced format. First, the \hat{f}_{Lz}^{pf} value for the motionless areas is computed using the field average and a Lanczos kernel. That is,

$$\hat{f}_{Lz}^{pf}(i + \Delta_i, j + \Delta_j, n) = \sum_{k=-b+1}^b h_{Lz}(\Delta_i' - k, b) \cdot (\hat{f}_{Lz}^{if}(i + k, j + \Delta_j, n) \cdot \delta(\lfloor 2 \cdot \Delta_i \rfloor) + \hat{f}_{FA}^{if}(i + k, j + \Delta_j, n) \cdot \delta(1 - \lfloor 2 \cdot \Delta_i \rfloor)), \quad (5)$$

where

$$\hat{f}_{FA}^{if}(i + k, j + \Delta_j, n) = \left(\hat{f}_{Lz}^{if}(i + k, j + \Delta_j, n - 1) + \hat{f}_{Lz}^{if}(i + k, j + \Delta_j, n + 1) \right) / 2, \quad (6)$$

where \hat{f}_{FA}^{if} represents the results of the field average. b represents the vertical size of the Lanczos kernel and δ denotes the Dirac delta function. Since the \hat{f}_{FA}^{if} value is at the 0.5 position in the vertical direction, Δ_i for the sub-pixel position of the Lanczos kernel is modified:

$$\Delta_i' = 2 \cdot \Delta_i - \lfloor 2 \cdot \Delta_i \rfloor. \quad (7)$$

The value of spatio-temporal interpolation, \hat{f}_{ST}^{pf} , is reconstructed with spatial and temporal filters. In order to improve the performance and to obtain the interpolated value at an arbitrary position, the spatial filter is defined as the Lanczos kernel. The temporal filter is determined to estimate the high-frequency components of the previous and next fields. Therefore, the \hat{f}_{ST}^{pf} value provides enhanced results using additional high-frequency components of temporal information. The \hat{f}_{ST}^{pf} is calculated by

$$\hat{f}_{ST}^{pf}(i + \Delta_i, j + \Delta_j, n) = \sum_{k=-b+1}^b h_{Lz}(\Delta_i - k, b) \cdot \hat{f}_{Lz}^{if} \times (i + k, j + \Delta_j, n) + \text{THF}, \quad (8)$$

where

$$THF = \begin{cases} THF_{i-1} \cdot (0.5 - \Delta_i) + THF_i \cdot (0.5 + \Delta_i) & \text{if top fields and } \Delta_i \leq 0.5 \\ THF_i \cdot (1.5 - \Delta_i) + THF_{i+1} \cdot (\Delta_i - 0.5) & \text{if top fields and } \Delta_i > 0.5 \\ THF_i \cdot (0.5 + \Delta_i) + THF_{i+1} \cdot (0.5 - \Delta_i) & \text{if bottom fields and } \Delta_i \leq 0.5 \\ THF_{i+1} \cdot (\Delta_i - 0.5) + THF_i \cdot (1.5 - \Delta_i) & \text{if bottom fields and } \Delta_i > 0.5 \end{cases} \quad (9)$$

where THF represents the high-frequency components of temporal information. Since the THF value is estimated only at the existing pixel position in the previous and next fields, a linear combination between the THFs at the neighboring pixels is used to obtain THF at an arbitrary position. According to characteristics of the top and bottom fields, the used THF is determined as shown in (9). THF_{*i*} is defined as

$$THF_i = \sum_{k=-c}^c h_{TF}(c+k) \cdot \left(\hat{f}_{Lz}^{if}(i+k, j+\Delta_j, n-1) + \hat{f}_{Lz}^{if}(i+k, j+\Delta_j, n+1) \right) / 2, \quad (10)$$

where

$$h_{TF} = [-0.25 \quad 0.5 \quad -0.25]. \quad (11)$$

Figure 5 shows an example of obtaining the high-frequency component of temporal information.

For combining the results of Lanczos interpolation and spatio-temporal interpolation, the reliability terms of temporal information are used as weighting factors based on both motion detection and feathering artifacts. The motion detection process and the feathering artifact detection process are performed in each pixel position. For motion detection, the temporal difference, D_T is computed through five fields in order to detect both normal and fast motions. D_T is given by

$$D_T = |\hat{f}_{Lz}^{if}(i+k, j+\Delta_j, n-1) - \hat{f}_{Lz}^{if}(i+k, j+\Delta_j, n+1)| + \sum_{k=0}^1 \sum_{l=0,2} |\hat{f}_{Lz}^{if}(i+k, j+\Delta_j, n-2+l) - \hat{f}_{Lz}^{if}(i+k, j+\Delta_j, n+l)| / 2. \quad (12)$$

$|x|$ returns the absolute value of x . To detect the feathering artifacts, the vertical difference, D_V is computed by using \hat{f}_{FA}^{if} .

$$D_V = \min\{D_{V1}, D_{V2}, D_{V3}\}, \quad (13)$$

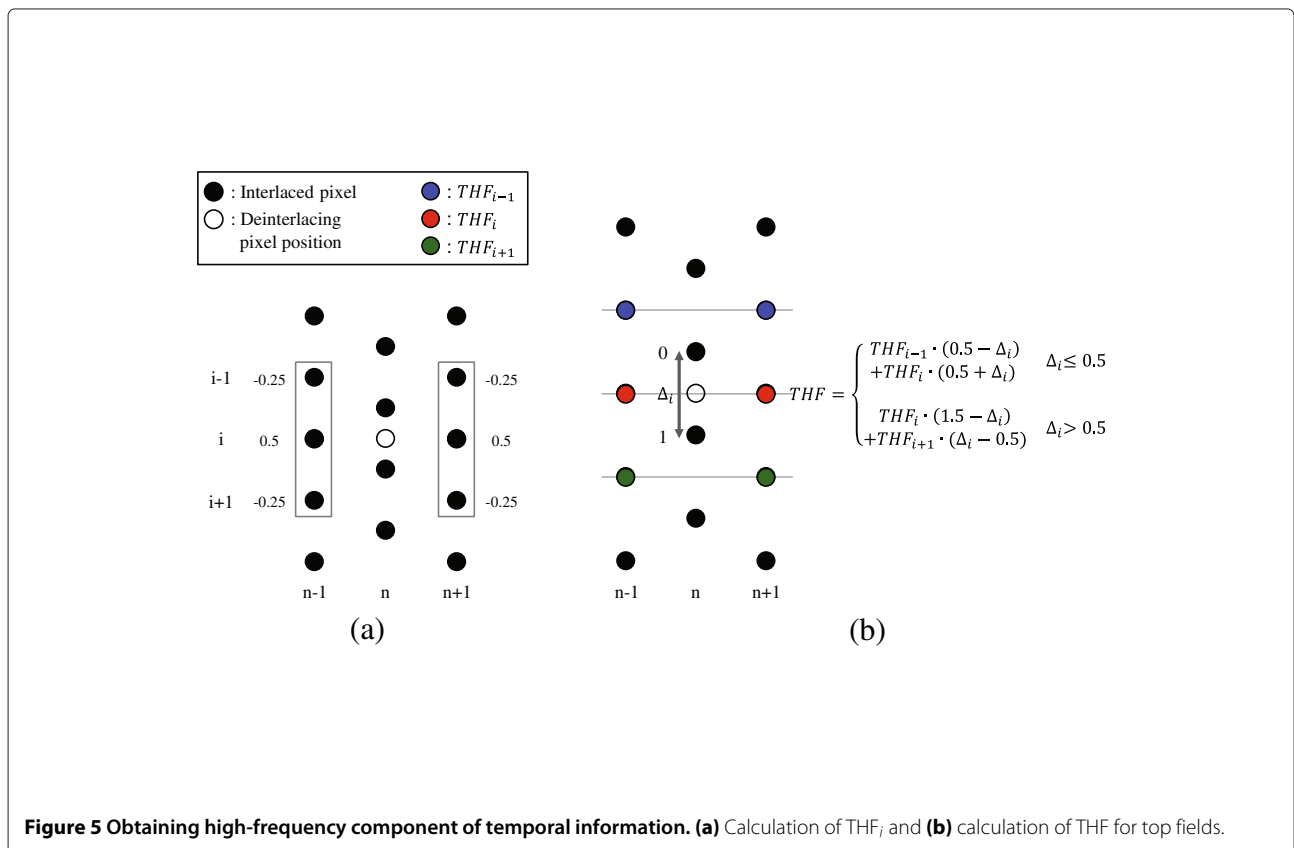


Figure 5 Obtaining high-frequency component of temporal information. (a) Calculation of THF_{*i*} and (b) calculation of THF for top fields.

where

$$\begin{aligned} D_{V1} &= |\hat{f}_{Lz}^{if}(i, j + \Delta_j, n) - \hat{f}_{FA}^{if}(i, j + \Delta_j, n)| \\ D_{V2} &= |\hat{f}_{Lz}^{if}(i, j + \Delta_j, n) - \hat{f}_{FA}^{if}(i - 1, j + \Delta_j, n)| \\ D_{V3} &= |\hat{f}_{Lz}^{if}(i + 1, j + \Delta_j, n) - \hat{f}_{FA}^{if}(i + 1, j + \Delta_j, n)|, \end{aligned} \quad (14)$$

The arbitration rules for α in (3) can be summarized as

$$\alpha = \frac{\min\{D_T + D_V, \tau_1\}}{\tau_1}, \quad (15)$$

where τ_1 represents a predetermined constant for normalization. According to (12), if D_T has a large value, the current pixel can be considered a motion pixel. In these motion areas, D_V has a large value generally because of the feathering artifacts. Thus, the final result is close to the result of spatio-temporal interpolation.

3.1.2 Ringing artifact reduction technique to improve interpolation performance

As discussed in Section 2, the side lobes of a Lanczos kernel can improve reconstruction performance along edge regions, but they often degrade image quality due to ringing artifacts. Moreover, although the temporal high pass filter of spatio-temporal interpolation can be useful to improve edge information, the resulting images generally suffer from shooting artifacts caused by overshooting or undershooting. In our previous work [18], we introduced a kernel-based image upscaling method that handled the ringing artifact problem caused by using a wider window for the Lanczos kernel. An extension of these concepts is now presented in order to avoid ringing artifacts or shooting artifacts. The proposed method is performed one-dimensionally after performing each one-dimensional interpolation process including Lanczos interpolation and spatio-temporal interpolation. Since ringing artifact reduction processes for Lanczos interpolation are similar to those discussed in [18], we only describe the ringing artifact reduction process for spatio-temporal interpolation in this section.

The proposed ringing artifact reduction method is composed of two steps: median filtering and arbitration. First, a median filter is applied to the result of spatio-temporal interpolation with the two nearest input data values. The result of median filtering is called the median spatio-temporal value in this paper. The median spatio-temporal value of the vertical direction is obtained as

$$\begin{aligned} \hat{f}_{medST}^{pf}(i + \Delta_i, j + \Delta_j, n) &= \text{med} \left\{ \hat{f}_{arbiLz}^{if}(i, j + \Delta_j, n), \right. \\ &\left. \hat{f}_{ST}^{pf}(i + \Delta_i, j + \Delta_j, n), \hat{f}_{arbiLz}^{if}(i + 1, j + \Delta_j, n) \right\}, \end{aligned} \quad (16)$$

where \hat{f}_{medST}^{pf} and \hat{f}_{arbiLz}^{if} represent the result of the median spatio-temporal value and the ringing artifact reduced result of the horizontal Lanczos interpolator, respectively, and $\text{med}\{x, y, z\}$ represents a three-input median filter that

returns the median value of x , y , and z . This median process is very efficient at removing ringing artifacts since it is particularly good for removing shot noise. However, this process often degrades image details by restricting the interpolated value within the values of its neighboring pixels. This restriction prevents reconstructing the high-frequency components of an image.

In the second arbitration step, the two highly complementary results are combined effectively in order to avoid ringing artifacts while reconstructing the high-frequency components. The arbitration process is performed using the results of (8) and (16) as follows:

$$\begin{aligned} \hat{f}_{arbiST}^{pf}(i + \Delta_i, j + \Delta_j, n) &= \beta \cdot \hat{f}_{ST}^{pf}(i + \Delta_i, j + \Delta_j, n) \\ &+ (1 - \beta) \cdot \hat{f}_{medST}^{pf}(i + \Delta_i, j + \Delta_j, n), \end{aligned} \quad (17)$$

where β ($0 \leq \beta \leq 1$) represents the weighting coefficient that controls the contribution of the two results. In the edge regions, β is determined near 1 to reconstruct the high-frequency components by adopting the spatio-temporal result. However, in regions with ringing artifacts, β decreases to remove the artifacts by adopting the median spatio-temporal result. The arbitration weight β is obtained with the difference values between the neighboring pixels as

$$\beta = \frac{\min\{D_U, D_D, \tau_2\}}{\tau_2}, \quad (18)$$

where τ_2 represents a predetermined constant for normalization, and the differences (D_U and D_D) are obtained as

$$\begin{aligned} D_U &= \max \left\{ |\hat{f}_{arbiLz}^{if}(i - 1, j + \Delta_j, n) - \hat{f}_{arbiLz}^{if}(i, j + \Delta_j, n)|, \right. \\ &\left. |\hat{f}_{arbiLz}^{if}(i, j + \Delta_j, n) - \hat{f}_{arbiLz}^{if}(i + 1, j + \Delta_j, n)| \right\}, \\ D_D &= \max \left\{ |\hat{f}_{arbiLz}^{if}(i, j + \Delta_j, n) - \hat{f}_{arbiLz}^{if}(i + 1, j + \Delta_j, n)|, \right. \\ &\left. |\hat{f}_{arbiLz}^{if}(i + 1, j + \Delta_j, n) - \hat{f}_{arbiLz}^{if}(i + 2, j + \Delta_j, n)| \right\}, \end{aligned} \quad (19)$$

where D_U and D_D represent the difference values of the upper and lower sides, respectively. According to (18) and (19), if both D_U and D_D have large values, high-frequency edges exist in the current pixels. Thus, the final result is generally close to the result of the spatio-temporal value in order to preserve the high-frequency edges. However, if either D_U or D_D is a small value, the region is classified as either a flat or a step edge region. Therefore, the result of the median spatio-temporal value dominates the final decision of whether or not to remove the ringing artifacts.

In summary, the ringing artifact reduction process is performed in the horizontal direction with $\hat{f}_{Lz}^{if}(i + k, j + \Delta_j, n)$ in (2) and then the vertical ringing artifact reduction process with $\hat{f}_{arbiLz}^{if}(i + k, j + \Delta_j, n)$ is applied to the results

of both Lanczos interpolation and spatio-temporal interpolation. Thus, the final result of the initial resampling step is obtained by

$$\hat{f}_{\text{Init}}^{\text{pf}}(i + \Delta_i, j + \Delta_j, n) = \alpha \cdot \hat{f}_{\text{arbiLz}}^{\text{pf}}(i + \Delta_i, j + \Delta_j, n) + (1 - \alpha) \cdot \hat{f}_{\text{arbiST}}^{\text{pf}}(i + \Delta_i, j + \Delta_j, n). \quad (20)$$

3.2 Edge enhancement step

Based on the LFI approach, the proposed method in Section 3.1 is used to resize images to fit the display format. However, LFI approaches are likely to smooth image details during the resampling process, and they usually produce jagged edge artifacts in the diagonal edge regions. In this section, we propose an edge enhancement algorithm that corrects the jagged edge artifacts and improves the sharpness of the initially interpolated images. For convenience of notation, the initially upscaled image is represented as $F(i, j)$ instead of $\hat{f}_{\text{Init}}^{\text{pf}}(i, j, n)$ as is used in (20).

3.2.1 Jagged edge correction with an ellipsoidal kernel

In order to remove the jagged edge artifacts, an estimation of the edge direction is important. In [18-20], it is assumed that the edge direction of interest $F(i, j)$, where F represents the initially upscaled image, is piecewise constant and the gradient vectors within a small mask should on average be orthogonal to the edge direction. Therefore, the estimation of edge direction can be formulated as the task of finding a unit vector \mathbf{d} to minimize the following cost function:

$$\text{cost}(\mathbf{d}) = \sum_{k=-k_0}^{k_0} \sum_{l=-l_0}^{l_0} \left\{ \mathbf{d}^T \cdot \mathbf{g}(k, l) \right\}^2 = \mathbf{d}^T \cdot \sum_{k=-k_0}^{k_0} \sum_{l=-l_0}^{l_0} \left\{ \mathbf{g}(k, l) \cdot \mathbf{g}^T(k, l) \right\} \cdot \mathbf{d} = \mathbf{d}^T \mathbf{C} \mathbf{d}, \quad (21)$$

where $\mathbf{g}(k, l) = [g_v(k, l) \quad g_h(k, l)]^T$ represents a gradient vector of $F(i + k, j + l)$, and a covariance matrix \mathbf{C} is determined as

$$\mathbf{C} = \begin{bmatrix} \sum_k \sum_l g_v(k, l) g_v(k, l) & \sum_k \sum_l g_v(k, l) g_h(k, l) \\ \sum_k \sum_l g_v(k, l) g_h(k, l) & \sum_k \sum_l g_h(k, l) g_h(k, l) \end{bmatrix} = \begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{bmatrix}, \quad (22)$$

where g_v and g_h represent derivatives in the vertical and horizontal directions, respectively.

A unique vector \mathbf{d} that minimizes the cost function in (21) is a good estimate of an edge direction so that the JEC can be carried out in the direction of \mathbf{d} . However, this requires an additional technique such as singular value decomposition (SVD) to find a unique optimal solution

for \mathbf{d} [19]. Therefore, instead of finding the optimal solution, the cost function in (21) is used to estimate which vectors are closer to the edge direction in the proposed JEC method. Let $\mathbf{d} = [k \quad l]^T$ denote a vector pointing from the current pixel in (i, j) to the neighboring pixel in $(i + k, j + l)$. Then, the similarity between \mathbf{d} and the edge direction is obtained by the cost function in (21) as follows:

$$\text{cost}(\mathbf{d}) = [k \quad l] \cdot \begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{bmatrix} \cdot \begin{bmatrix} k \\ l \end{bmatrix} = c_{00}k^2 + (c_{01} + c_{10})kl + c_{11}l^2. \quad (23)$$

As mentioned above, $\text{cost}(\mathbf{d})$ decreases as the orientation of \mathbf{d} approaches the edge direction. However, if \mathbf{d} is parallel to the gradient vector, $\text{cost}(\mathbf{d})$ returns a large value. Using this characteristic, an adaptive smoothing kernel is obtained by adopting a Gaussian kernel as

$$\epsilon(k, l) = \exp \left\{ -\frac{\text{cost}(\mathbf{d})}{\sigma} \right\} = \exp \left\{ -\frac{\text{cost}([k \quad l]^T)}{\sigma} \right\}, \quad (24)$$

where $\epsilon(k, l)$ represents the coefficients of the adaptive smoothing kernel. The cost function in (23) represents an elliptic equation form, and it spread the Gaussian kernel along the local edge direction. Therefore, in this paper, this adaptive kernel is called an ellipsoidal kernel, which is similar to the steering kernel used in [21]. σ in (24) controls the scale of the kernel, and it is determined as

$$\sigma = \nu \cdot \max(c_{00}, c_{11}), \quad (25)$$

where ν represents a predetermined smoothing parameter. In Figure 6, an example of an ellipsoidal kernel is illustrated with the gradient vector, the edge direction, and an arbitrary vector \mathbf{d} . As shown in (24) and Figure 6, the ellipsoidal kernel assigns large weights along the edge direction. Therefore, pixels in the similar edge directions are smoothed by the ellipsoidal kernel, and as a result, jagged edge artifacts are corrected.

Figure 7 illustrates the proposed method performed with an ellipsoidal kernel. This figure shows the positions of the pixels used in the proposed method. In our experiments, a sufficient window size is necessary to ensure performance, but this also increases the computational cost. Therefore, we divide the proposed method into vertical and horizontal processes and use only the pixels within the cross-shaped region for the filtering process, as shown in Figure 7. Also, some pixels such as the four nearest neighbors of the current pixel degrade the correction performance since large weights are generally assigned to the pixels because of their short distances from the current pixel. Therefore, the vertical and horizontal lines that pass the current pixel are excluded from the filtering process,

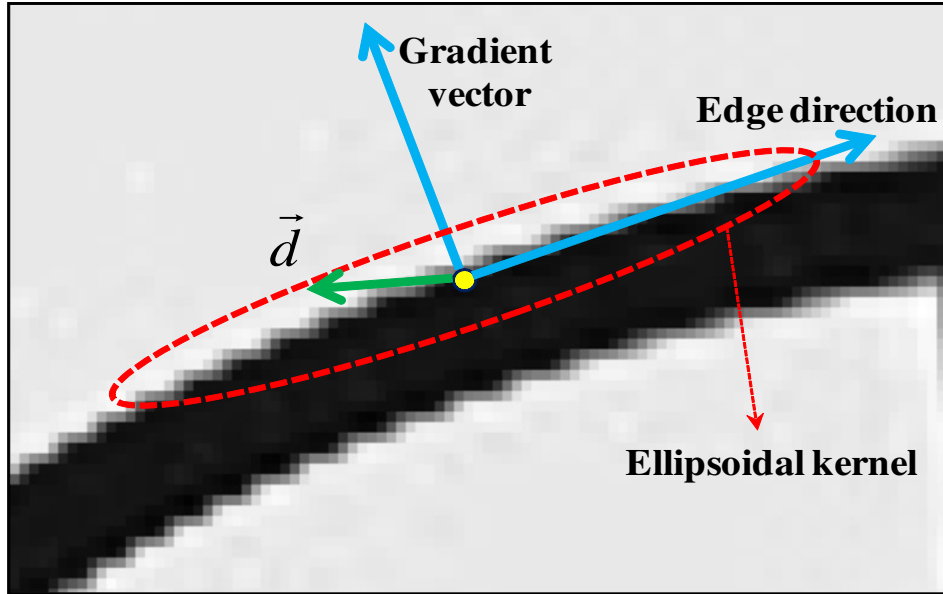


Figure 6 An ellipsoid kernel and its relation between the gradient vector and the edge direction.

as shown in Figure 7. With the remaining pixels, the proposed method is performed in the horizontal and vertical directions. First, the horizontal process is performed as

$$\begin{aligned}
 Z_{\text{hor}} &= \sum_{l \neq 0, l=-L}^L \epsilon(-1, l) + \epsilon(1, -l) = 2 \cdot \sum_{l \neq 0, l=-L}^L \epsilon(-1, l) \\
 F_{\text{Jc-hor}}(i, j) &= \sum_{l \neq 0, l=-L}^L \epsilon(-1, l) \cdot F(i-1, j+l) + \epsilon(1, -l) \cdot \\
 &\quad \times F(i+1, j-l) \\
 &= 2 \cdot \sum_{l \neq 0, l=-L}^L \epsilon(-1, l) \cdot F_{\text{hor}}(i, j+l),
 \end{aligned} \tag{26}$$

where F and $F_{\text{Jc-hor}}$ represent the initially interpolated image and the weighted sum of the horizontal process, respectively. Z_{hor} is used for normalization and $F_{\text{hor}}(i, j+l)$ is defined as

$$F_{\text{hor}}(i, j+l) = \frac{F(i-1, j+l) + F(i+1, j-l)}{2}. \tag{27}$$

In (26), $\epsilon(1, -l)$ is the same as $\epsilon(-1, l)$ since an ellipsoidal kernel is point symmetric with respect to the middle point. In the same way, the vertical process is performed as

$$Z_{\text{ver}} = 2 \cdot \sum_{|k|>1, k=-K}^K \epsilon(-k, -1),$$

$$F_{\text{Jc-ver}}(i, j) = 2 \cdot \sum_{|k|>1, k=-K}^K \epsilon(-k, -1) \cdot F_{\text{ver}}(i-k, j), \tag{28}$$

where $F_{\text{Jc-ver}}$ represents the weighted sum in the vertical direction. Z_{ver} is used for normalization and $F_{\text{ver}}(i-k, j)$ is defined as

$$F_{\text{ver}}(i-k, j) = \frac{F(i-k, j-1) + F(i+k, j+1)}{2}. \tag{29}$$

From (26) and (28), the final JEC result is obtained as

$$F_{\text{Jc}}(i, j) = \frac{F(i, j) + F_{\text{Jc-hor}}(i, j) + F_{\text{Jc-ver}}(i, j)}{1 + Z_{\text{hor}} + Z_{\text{ver}}}. \tag{30}$$

3.2.2 Jagged edge correction with transient improvement

In general, enlarging images result in blurring effects within initially interpolated images, and these blurring effects are perceived more conspicuously in regions with jagged edge artifacts. In these regions, the interpolation process was performed across the contrasting edges so that it produced blurred edges which appear like a staircase. In the proposed JEC method, an ellipsoidal kernel

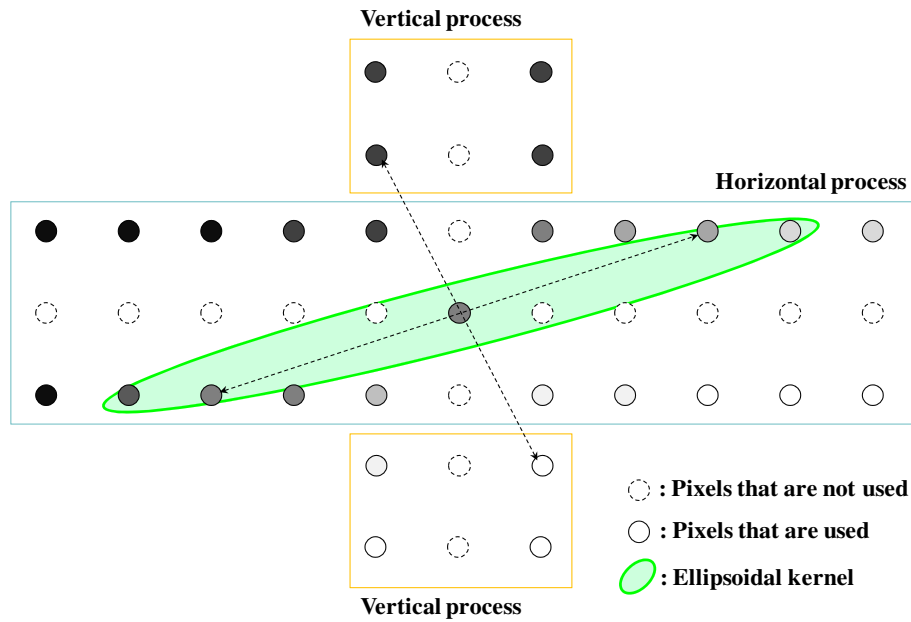


Figure 7 The proposed JEC process with an ellipsoidal kernel.

was used to reduce these stair-like artifacts by smoothing them along the local edge direction. However, since the ellipsoidal kernel is a kind of low pass filter, some image details are smoothed by the kernel during the filtering process. Therefore, in this paper, a TI technique is simultaneously performed with the filtering process in order to improve the sharpness of the initially interpolated images.

From the viewpoint of enhancing the sharpness of an image, the basic idea of the TI methods may seem similar to that of general sharpening methods. However, TI methods are more specialized in terms of improving the slow transitions of blurred edges since these methods fundamentally prevent overshooting and undershooting. In order to improve the slow transition with general sharpening methods, we have to increase the amount of sharpening. However, an excessive amount of sharpening tends to degrade image quality because it produces severe overshooting and undershooting. These factors often produce white and black bands along the contrasting edges, which appear unpleasant. However, TI algorithms can produce steep and natural edge transitions without undershooting and overshooting. In image upscaling applications, a lot of edges show poor transitions so the TI technique can be used efficiently to improve the sharpness of the upscaled images. In Figure 8, the behavior of the TI algorithm is illustrated briefly.

TI methods generally consist of two steps [22-24]. In the first step, a high-frequency boost filter is adopted as a pre-filter to enhance the slow transition of blurred edges. That is, a correction signal is added to the blurred signal to reconstruct the original high-frequency component. The above description can be written as the following equation:

$$X_{HB} = X_{in} + h_{HF} * X_{in} = h_{HB} * X_{in}, \quad (31)$$

where X_{in} and X_{HB} represent the input signal and the filtered result, respectively. h_{HF} and h_{HB} represent a high pass filter and a high-frequency boost filter, respectively. For example, the second-order derivative operator has been used as h_{HF} in several conventional methods. In the second step, the processed signal is limited to the proper range to prevent overshooting and undershooting:

$$X_{TI}(i, j) = TI \{ X_{HB}(i, j), \tau_{max}, \tau_{min} \} = \begin{cases} \tau_{max}, & \text{if } X_{HB}(i, j) > \tau_{max} \\ \tau_{min}, & \text{if } X_{HB}(i, j) < \tau_{min} \\ X_{HB}(i, j), & \text{otherwise} \end{cases} \quad (32)$$

In most TI algorithms, τ_{max} and τ_{min} are usually set as the local maximum and minimum values found within a predefined window. In this section, the TI method is combined with the filtering process, described in

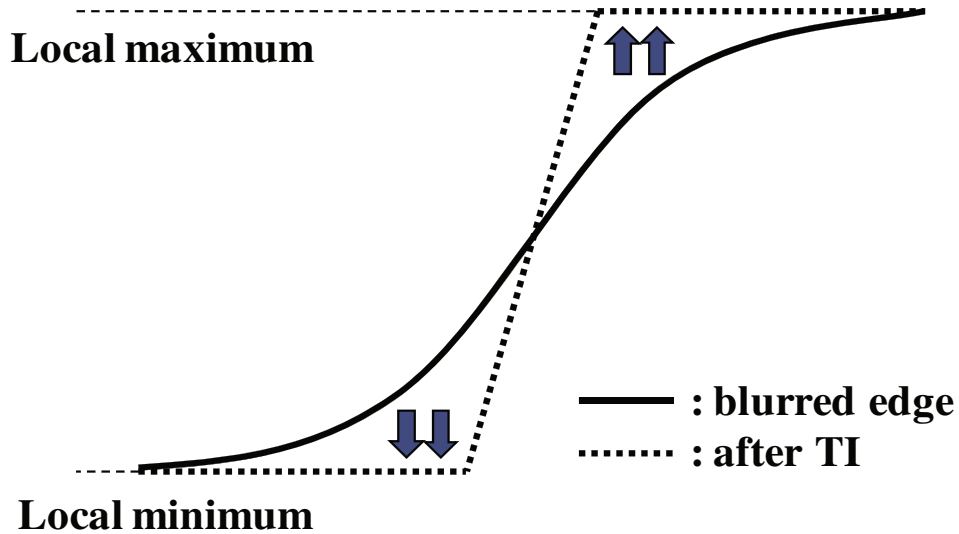


Figure 8 Transient improvement method using local maximum and local minimum values.

Section 3.2.1. The TI process is performed in the vertical and horizontal directions. Since the vertical and horizontal filtering processes are exactly the same, we will only describe the horizontal process in this paper.

In (26) and (27), $F_{\text{hor}}(i, j + l)$ is used for the filtering process, and it is obtained by averaging $F(i - 1, j + l)$ and $F(i + 1, j - l)$. However, this direct average causes blurring effects during the filtering process when the two values are quite different. Therefore, the TI technique is applied to $F_{\text{hor}}(i, j + l)$ to reduce the blurring effects. Let

$F_{\text{TI-hor}}(i, j + l)$ be the result of the TI process, which is used instead of $F_{\text{hor}}(i, j + l)$. Then, (26) is changed to the following equation:

$$Z_{\text{hor}} = 2 \cdot \sum_{l \neq 0, l = -L}^L \epsilon(-1, l) \tag{33}$$

$$F_{\text{Jc-hor}}(i, j) = 2 \cdot \sum_{l \neq 0, l = -L}^L \epsilon(-1, l) \cdot F_{\text{TI-hor}}(i, j + l),$$

$$TI_{\text{out}} = TI(0.5a + 0.5b_1 + 0.5b_2 - 0.25c_1 - 0.25c_2, \tau_{\text{max}}, \tau_{\text{min}})$$

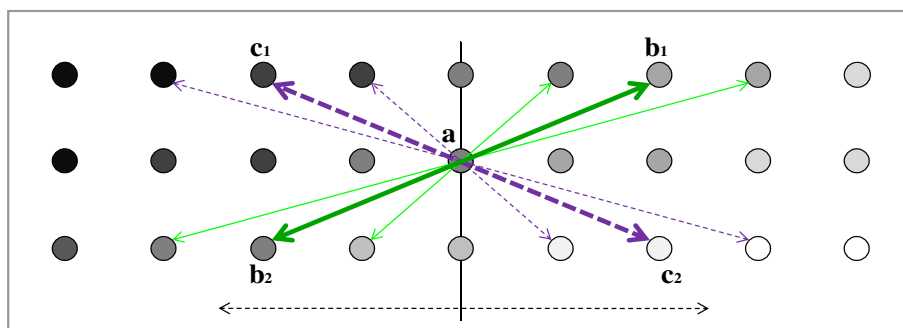


Figure 9 JEC method with the TI process and positions of the pixels used for the TI process.

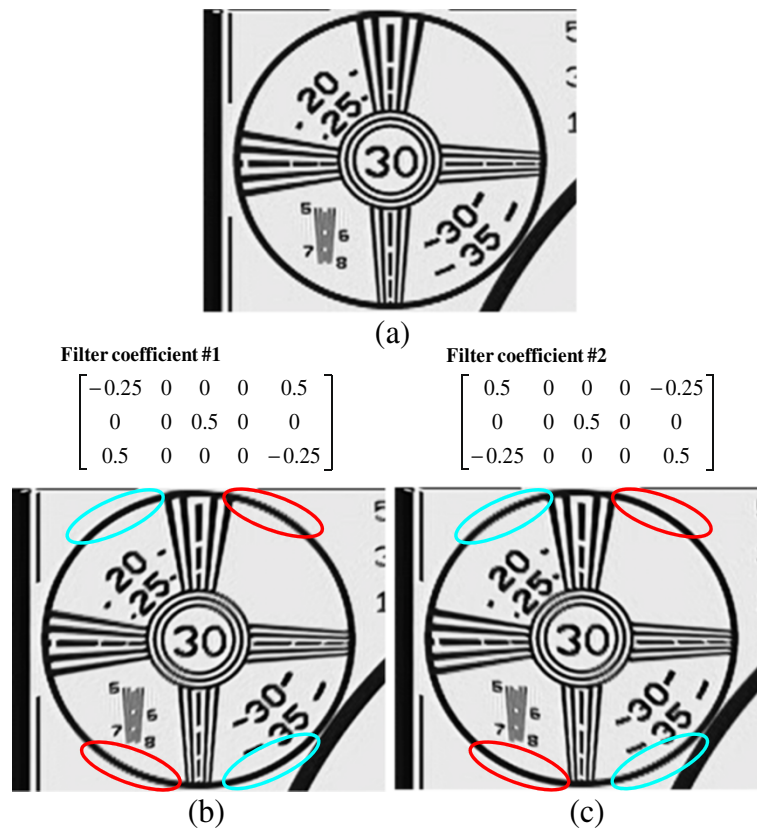


Figure 10 The effect of the proposed TI process performed in two directions. (a) Upscaled image produced by the Lanczos interpolator with the ringing reduction and the results of the TI process when (b) $l = 2$ and (c) $l = -2$.

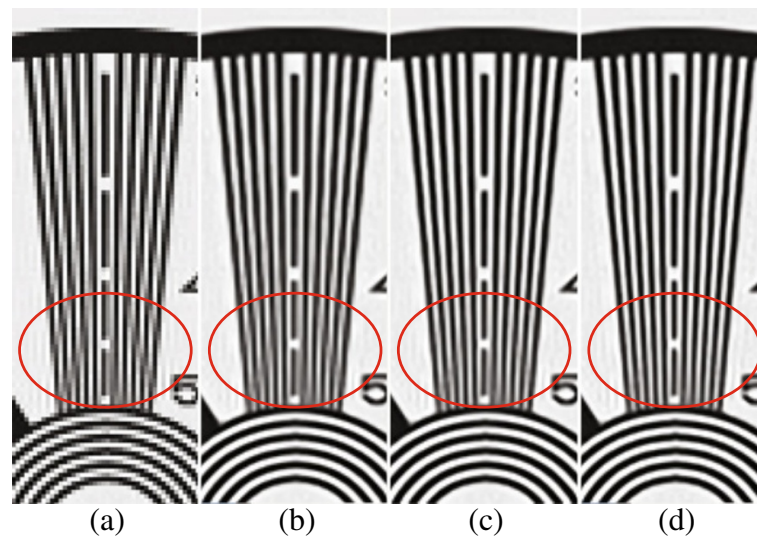


Figure 11 Upscaled images obtained by varying the size of the Lanczos kernel. (a) Input image. Horizontal kernel size (b) $a = 4$, (c) $a = 8$, and (d) $a = 12$.

where $F_{TI\text{-hor}}(i, j + l)$ is obtained as

$$F_{TI\text{-hor}}(i, j + l) = TI \{ F_{HB\text{-hor}}(i, j + l), \tau_{\max}, \tau_{\min} \}, \quad (34)$$

where $F_{HB\text{-hor}}(i, j + l)$ represents the result of high-frequency boost filtering. It is obtained by adding the high-frequency components to $F_{hor}(i, j + l)$ as

$$\begin{aligned} F_{HB\text{-hor}}(i, j + l) &= F_{hor}(i, j + l) + \begin{bmatrix} 0.5 \\ -0.25 \\ -0.25 \end{bmatrix}^T \cdot \begin{bmatrix} F(i, j) \\ F(i - 1, j - l) \\ F(i + 1, j + l) \end{bmatrix} \\ &= \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \\ -0.25 \\ -0.25 \end{bmatrix}^T \cdot \begin{bmatrix} F(i - 1, j + l) \\ F(i + 1, j - l) \\ F(i, j) \\ F(i - 1, j - l) \\ F(i + 1, j + l) \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \\ -0.25 \\ -0.25 \end{bmatrix}^T \cdot \begin{bmatrix} b_1 \\ b_2 \\ a \\ c_1 \\ c_2 \end{bmatrix}, \end{aligned} \quad (35)$$

where l falls within the range of $[-L, L]$. The positions of the pixels $[b_1, b_2, a, c_1, c_2]$ are illustrated in Figure 9 as an example.

As shown in Figure 9, $F_{hor}(i, j + l)$ is obtained by averaging b_1 and b_2 in the direction of the green arrow. High pass

filtering is applied to the pixels $[a, c_1, c_2]$ in the opposite direction to $F_{hor}(i, j + l)$ (along the dotted purple arrow). We supposed the direction of the green arrow to be similar to the real edge direction. Then, $a, c_1,$ and c_2 in the opposite direction to the green arrow form a contrasting edge, and the high-frequency components are extracted from this edge. That is, the high-frequency components obtained from pixels across an edge ($a, c_1,$ and c_2) are used to restore the blurred edges of an initially interpolated image. After the filtering process, the filtered value is limited to the proper range between τ_{\max} and τ_{\min} to prevent artifacts caused by overshooting and undershooting:

$$\begin{bmatrix} \tau_{\max} \\ \tau_{\min} \end{bmatrix} = \begin{bmatrix} F_{\max}(l) \\ F_{\min}(l) \end{bmatrix} = \begin{bmatrix} \max_{n \in \{|n| - |l| \leq n \leq |l|\}} F(i, j + n) \\ \min_{n \in \{|n| - |l| \leq n \leq |l|\}} F(i, j + n) \end{bmatrix}, \quad (36)$$

where $|l|$ returns the absolute value of l . $F_{\max}(l)$ and $F_{\min}(l)$ represent the local maximum and local minimum values within the range of $[-|l|, |l|]$. The local extremums are searched on the middle line among the three lines illustrated in Figure 9. In general, the middle line usually

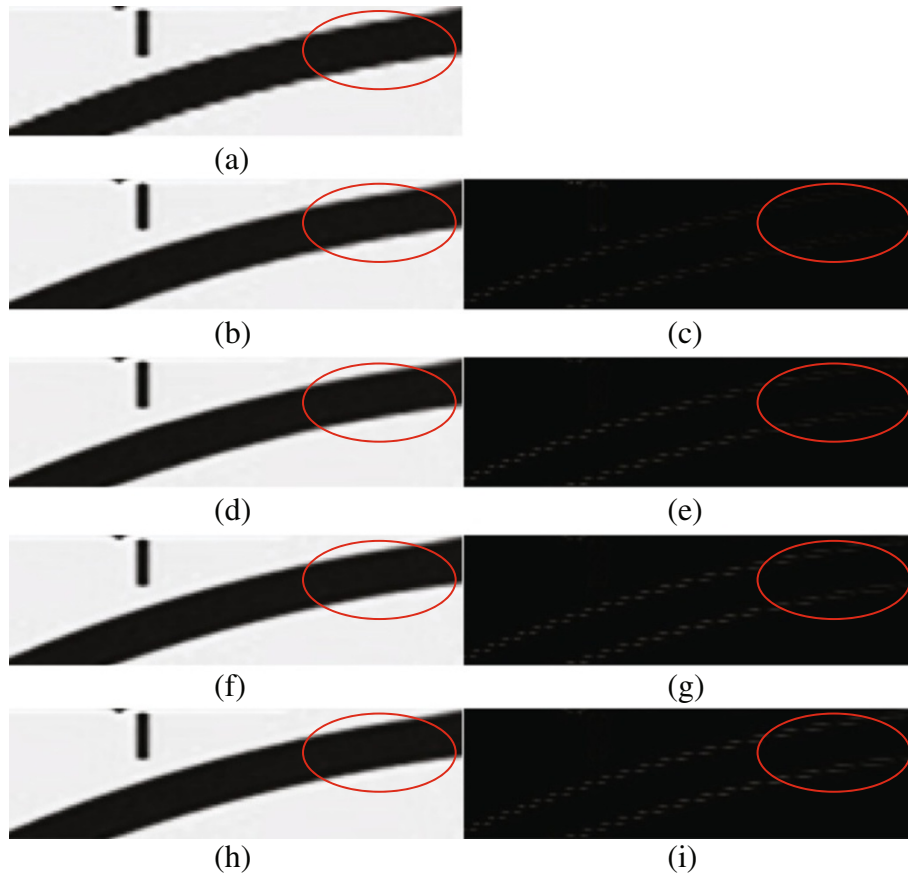


Figure 12 Jagged edge corrected images obtained by varying the window size of the JEC process. **(a)** Input image with jagged edge artifacts. Horizontal window size **(b)** $L = 5$, **(d)** $L = 7$, **(f)** $L = 9$, and **(h)** $L = 11$. The difference images between the input image and the reconstructed image are presented in **(c)**, **(e)**, **(g)**, and **(i)**.

crosses over the edges on (i, j) so that it contains both the maximum and minimum pixel values within a local window. From (34) to (36), we obtain $F_{TI-hor}(i, j + l)$, which provides sharper images than $F_{hor}(i, j + l)$.

In Figure 10, we present two examples of the proposed TI process when $l = 2$ and $l = -2$. As shown in Figure 10b, jagged edge artifacts are reduced along the edges at 0° to 45° angles (blue circled) when $l = 2$. However, when $l = -2$, the jagged edge artifacts are

reduced along the edges at 135° to 180° angles (red circled) in Figure 10c. Even though the edges at 135° to 180° angles in Figure 10b and the edges at 0° to 45° angles in Figure 10c are degraded by the TI process, the degraded results are excluded by the ellipsoidal kernel. In the proposed method, the ellipsoidal kernel assigns large weights along the estimated edge direction. Therefore, the $l = 2$ results in Figure 10b dominate the final results along the edges at 0° to 45° angles, and the $l = -2$ results in

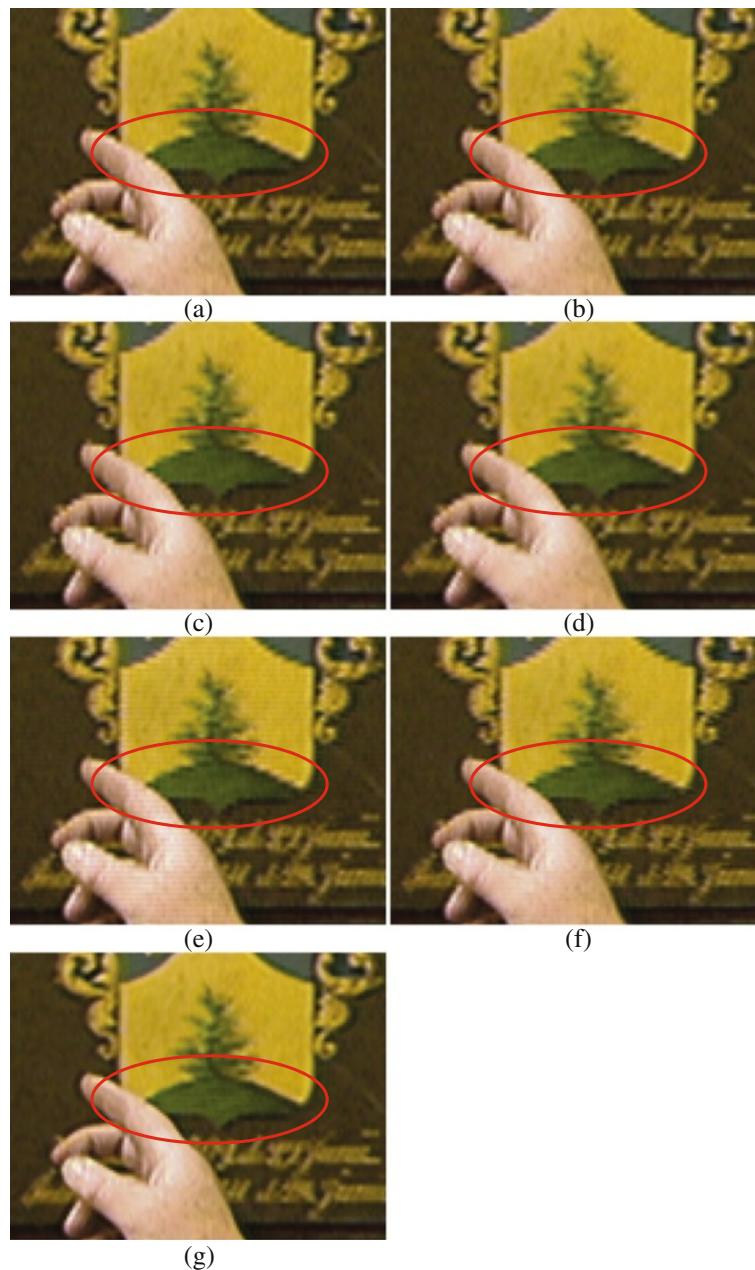


Figure 13 Results of conventional deinterlacing methods and the proposed method. (a) CS, (b) Lanczos, (c) NEDD, (d) LSMD, (e) STCAD, (f) MAVTF, and (g) the proposed method.

Table 1 PSNRs (dB) of various deinterlacing methods including the proposed method

	LFI		EDI		Combination		
	CS	Lanczos	NEDD	LSMD	STCAD	MAVTF	Proposed method
Bus	28.44	28.43	28.23	28.05	28.24	28.78	28.82
Coastguard	29.08	29.31	29.31	29.29	31.38	31.23	<i>31.77</i>
Container	28.96	29.08	29.00	28.84	35.55	35.45	<i>35.71</i>
Foreman	34.31	34.50	34.60	35.00	36.43	36.78	<i>37.51</i>
Hall monitor	32.10	32.20	32.07	31.65	38.25	37.75	<i>38.43</i>
Mom daughter	33.38	33.67	33.76	33.69	38.14	38.30	<i>38.48</i>
Silent	34.13	34.38	34.47	34.59	39.93	40.33	<i>40.43</i>
Rodmap	28.49	28.09	27.51	28.36	30.25	29.85	<i>30.47</i>
Washdc	32.49	32.23	31.80	31.82	38.14	37.70	<i>38.48</i>
Mobile	32.54	32.84	32.92	32.89	34.53	34.90	<i>34.41</i>
Stockholm	32.55	32.88	32.99	32.93	33.83	33.07	<i>33.85</i>
Shields	30.99	31.25	<i>31.31</i>	31.21	30.92	30.70	<i>31.17</i>
Average	31.45	31.57	31.49	31.52	34.63	34.57	<i>34.96</i>

The highest PSNR values are set in italics.

Figure 10c dominate the final results along the edges at 135° to 180° angles, respectively.

For the vertical process, $F_{TI-ver}(i-k, j)$ is used instead of $F_{ver}(i-k, j)$, and it is obtained by

$$F_{TI-ver}(i-k, j) = TI \{ F_{HB-ver}(i-k, j), \tau_{max}, \tau_{min} \}, \quad (37)$$

$$Z_{ver} = 2 \cdot \sum_{|k|>1, k=-K}^K \epsilon(-k, -1),$$

$$F_{Jc-ver}(i, j) = 2 \cdot \sum_{|k|>1, k=-K}^K \epsilon(-k, -1) \cdot F_{TI-ver}(i-k, j). \quad (38)$$

and the vertical JEC process is performed as

The final JEC result is obtained by (30).

Table 2 SSIMs of various deinterlacing methods including the proposed method

	LFI		EDI		Combination		
	CS	Lanczos	NEDD	LSMD	STCAD	MAVTF	Proposed method
Bus	0.912	0.913	0.908	0.904	0.910	0.915	<i>0.916</i>
Coastguard	0.831	0.838	0.839	0.835	<i>0.918</i>	0.916	0.914
Container	0.911	0.916	0.917	0.912	0.984	0.982	<i>0.985</i>
Foreman	0.940	0.944	0.947	0.948	0.958	0.960	<i>0.967</i>
Hall monitor	0.967	0.968	0.968	0.962	0.978	0.975	<i>0.980</i>
Mom daughter	0.946	0.949	0.952	0.950	0.969	0.972	<i>0.973</i>
Silent	0.927	0.931	0.932	0.933	0.972	<i>0.987</i>	0.985
Rodmap	0.938	0.936	0.931	0.940	0.962	0.964	<i>0.967</i>
Washdc	0.955	0.950	0.952	0.951	0.988	0.981	<i>0.988</i>
Mobile	0.908	0.916	0.918	0.921	0.932	0.936	<i>0.937</i>
Stockholm	0.890	0.898	0.901	0.900	0.913	0.915	<i>0.915</i>
Shields	0.913	0.918	0.920	0.918	0.921	0.916	<i>0.921</i>
Average	0.919	0.923	0.923	0.922	0.950	0.951	<i>0.954</i>

The highest SSIM values are set in italics.

4 Experimental results

The performance of the proposed method was tested with well-known common intermediate format (CIF), SD, and HD video sequences. We converted the progressive sequences into an interlaced format or down-sampled them according to experimental purposes. For performance comparisons, three groups of conventional deinterlacing methods were implemented: LFI approaches including cubic spline (CS) [11] and Lanczos, EDI approaches including new edge dependent deinterlacing (NEDD) [13] and local surface model-based deinterlacing (LSMD) [14], and a combination of both LFI and EDI approaches including spatial-temporal content-adaptive deinterlacing (STCAD) [3] and motion adaptive vertical temporal filtering (MAVFT) [4]. Two groups of image upscaling methods were implemented: LFI approaches including CS [11], EDI approaches including new edge-directed interpolation (NEDI) [15] and soft-decision adaptive interpolation (SAI) [17]. In the experiments, the

interpolated positions of deinterlacing and upscaling were adaptively adjusted for the CS and Lanczos, respectively. Lanczos represents Lanczos interpolation without applying the proposed ringing reduction method.

The peak signal-to-noise ratio (PSNR) and the structural similarity (SSIM) [25] of the luminance channel were used for quantitative measurement. The PSNR is defined as $PSNR = 10 \log_{10} (255^2 / MSE)$, where MSE represents the mean squared error between the original and the reconstructed images. The SSIM compared local patterns of pixel intensities that were normalized for luminance and contrast. Thus, the SSIM was used to gauge the visual quality of images more closely to the human visual system. The SSIM index is represented by

$$SSIM(o_m, r_m) = \frac{(2\mu_o\mu_r + C1)(2\sigma_{or} + C2)}{(\mu_o^2 + \mu_r^2 + C1)(\sigma_o^2 + \sigma_r^2 + C2)}, \quad (39)$$

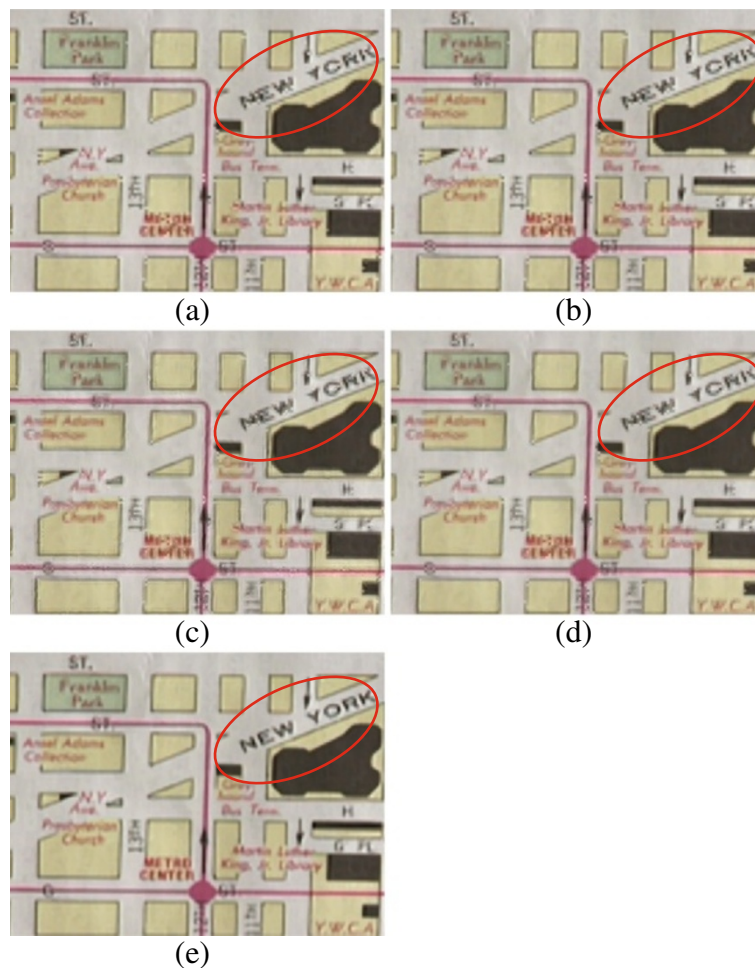


Figure 14 Results of conventional VDFC methods and the proposed method. (a) CS, (b) Lanczos, (c) NEDI, (d) SAI, and (e) the proposed method. Especially, resampled images of CS, Lanczos, NEDI, and SAI were obtained after performing the STCAD.

Table 3 PSNRs (dB) of various VDFC methods including the proposed method

	LFI		EDI		Combination
	CS	Lanczos	NEDI	SAI	Proposed method
Bus	21.11	21.35	20.84	21.22	21.43
Coastguard	24.18	24.36	24.05	24.27	25.40
Container	24.01	24.31	23.58	24.13	24.96
Foreman	30.52	30.96	30.43	30.77	31.33
Hall monitor	25.86	26.00	25.26	25.76	26.34
Mom daughter	29.14	29.62	28.82	29.25	30.11
Silent	29.77	30.50	29.65	29.98	30.32
Rodmap	20.58	20.83	20.38	20.65	21.25
Washdc	24.59	24.77	23.75	24.37	24.31
Mobile	27.54	27.81	27.02	27.57	27.88
Stockholm	27.79	28.06	27.42	27.85	28.22
Shields	25.45	25.82	25.28	25.62	25.93
Average	25.87	26.19	25.54	25.95	26.45

Resampled images of CS, Lanczos, NEDI, and SAI were obtained after performing the STCAD. The highest PSNR values are set in italics.

where o and r represent the original and reconstructed images, respectively. o_m and r_m are the image contents at the m th windows of the image. μ and σ represent the mean and standard deviation of each pixel over a 11×11 pixel Gaussian window, respectively. σ_{or} represents the covariance value of each pixel over a 11×11 pixel Gaussian window on the original and reconstructed images. $C1$ and $C2$ are constant values to increase stability: $C1 = 6.5025$ and $C2 = 58.5225$ for 8-bit images. In order to evaluate the overall image quality, a mean structural similarity (MSSIM) was employed.

$$MSSIM(o, r) = \frac{1}{M} \sum_{m=1}^M SSIM(o_m, r_m), \quad (40)$$

where M represents the number of local windows of the image. The MSSIM ranged from 0 to 1. Therefore, a higher MSSIM value close to 1 meant that a given image was reconstructed with reduced degradation of structural information.

There are several parameters in the proposed method. Most of these parameters were set empirically and tested with various images to obtain the best results. a in (2) and

Table 4 SSIMs of various VDFC methods including the proposed method

	LFI		EDI		Combination
	CS	Lanczos	NEDI	SAI	Proposed method
Bus	0.648	0.653	0.643	0.650	0.651
Coastguard	0.584	0.588	0.582	0.585	0.641
Container	0.773	0.778	0.769	0.775	0.824
Foreman	0.878	0.881	0.872	0.879	0.896
Hall monitor	0.860	0.864	0.843	0.862	0.877
Mom daughter	0.880	0.883	0.874	0.881	0.897
Silent	0.831	0.834	0.825	0.832	0.848
Rodmap	0.768	0.773	0.765	0.771	0.776
Washdc	0.803	0.809	0.794	0.805	0.808
Mobile	0.779	0.786	0.778	0.787	0.792
Stockholm	0.754	0.761	0.751	0.759	0.761
Shields	0.722	0.728	0.720	0.726	0.733
Average	0.773	0.778	0.768	0.776	0.792

Resampled images of CS, Lanczos, NEDI, and SAI were obtained after performing the STCAD. The highest SSIM values are set in italics.

b in (5) represent the sizes of the interpolation filters in the horizontal and vertical directions, respectively. Since the side lobes of a Lanczos kernel improved reconstruction performance along the edge regions, the used kernel size

severely impacted the performance of the reconstructed image. In Figure 11, we presented the upscaled images obtained by varying the size of the Lanczos interpolation kernel to analyze the effect of the kernel size on the

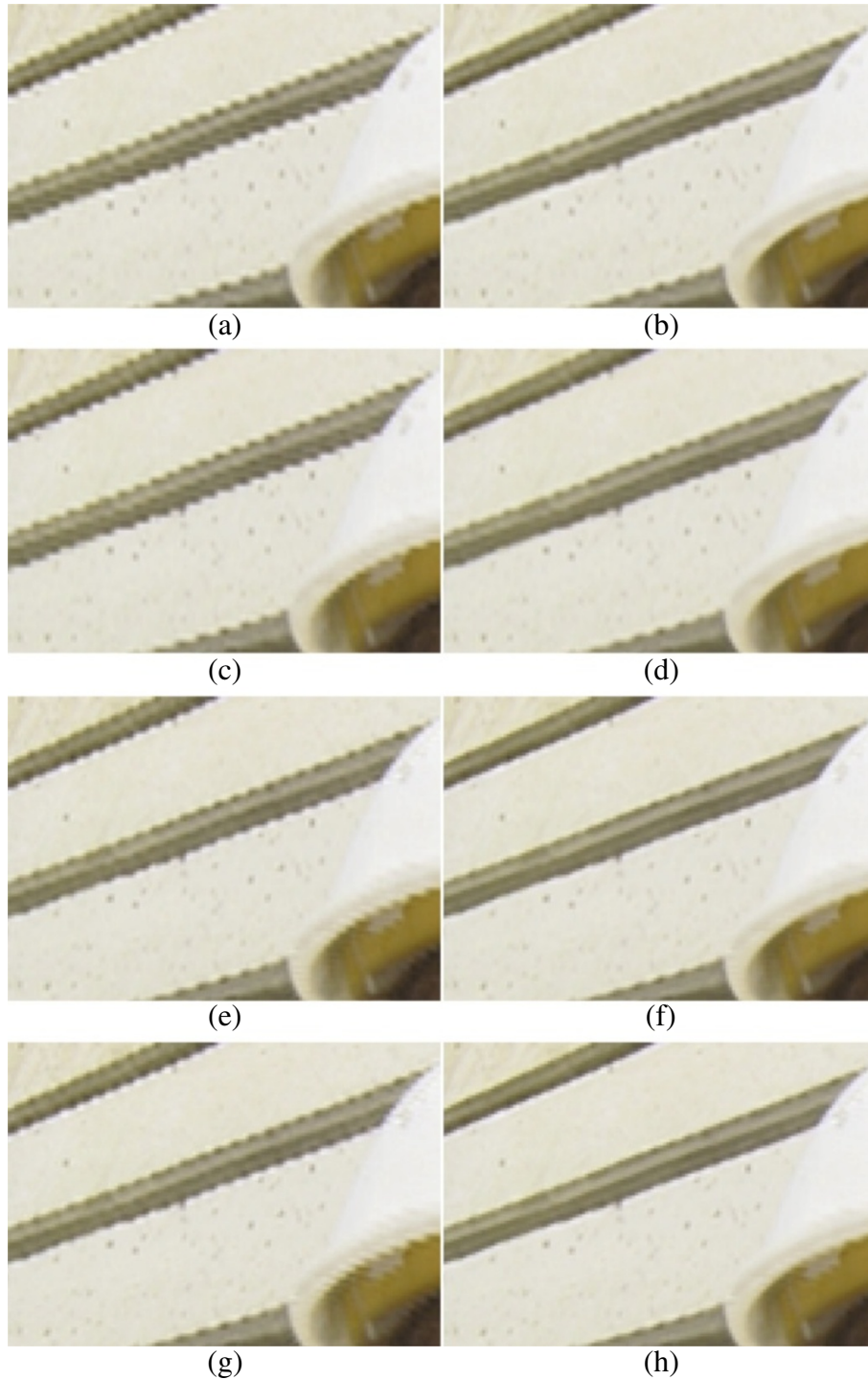


Figure 15 Results of the proposed JEC method after the deinterlacing process. (a) CS, (b) JEC after CS, (c) Lanczos, (d) JEC after Lanczos, (e) STCAD, (f) JEC after STCAD, (g) MAVTF, and (h) JEC after MAVTF.

image quality. Figure 11a presents the input image with the vertically patterned edges. In Figure 11b,c,d, it was upscaled by a factor of 2, and different kernel sizes were used for each result. The horizontal kernel sizes were set as 4, 8, and 12 in Figure 11b,c,d, respectively. As shown in the figures, the high-frequency components were reconstructed well as the kernel size increased. In our observations, there was little improvement when the size was larger than 8. Therefore, we set the size as 8, as a compromise between performance and hardware complexity. In the experiments, we used a smaller interpolation filter in the vertical direction because of the line memory restriction of the hardware structure. Thus, a and b were set as $a = 8$ and $b = 6$, respectively. Also, τ_1 in (15) and τ_2 in (18) represent the predetermined constants for normalization. For 8-bit images, τ_1 was set as 96 for the mixing process of vertical interpolation, and τ_2 was set as 64 for the ringing artifact reduction process. Generally, the window size used for the JEC process is the most important factor in determining performance, and therefore, a sufficient window size was required to reduce the jagged edge artifacts along the nearly horizontal or vertical edges. In Figure 12, we presented the jagged edge corrected images obtained by varying the window size of the JEC process to analyze the effect of the window size on the image quality. Figure 12a presents the input image

with the nearly horizontal edge, which was degraded by the jagged edge artifacts. In Figure 12b,d,f,h, it was corrected by the JEC process, and different window sizes were used for each result. The horizontal window size were set as 5, 7, 9, and 11, respectively. As shown in the figures, the jagged edge artifacts were reduced substantially by the JEC process. In order to compare the performance of the JEC process with various window sizes, the difference images between the input image and the reconstruction images are presented in Figure 12c,e,g,i, and the difference image presents the jagged edge artifacts corrected by the JEC process. As shown in the figures, the jagged edge artifacts were reduced well as the window size increased. In our observation, there was little improvement when the size was larger than 7. Therefore, L in (26) and (33) was set as 7 for the horizontal process. However, the vertical window size K in (28) and (38) was set as 3 to reduce the line memory required for hardware implementation. We used the same parameter values for all the test images.

In Figure 13, the performance of the proposed method was evaluated for the deinterlacing process. For this experiment, test video sequences were converted into an interlaced format and then the interlaced sequences were deinterlaced again with conventional deinterlacing methods. Figure 13a,b,c,d,e,f,g represents the deinterlaced results of CS, Lanczos, NEDD, LSMD, STCAD,

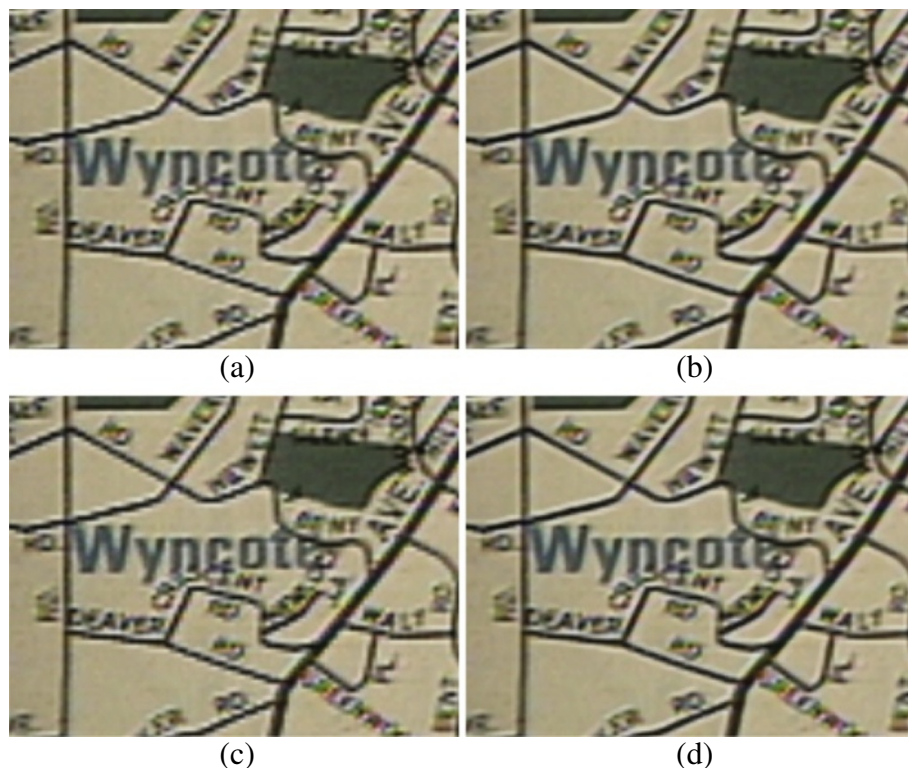


Figure 16 Results of the proposed JEC method after the image upscaling process. (a) CS, (b) JEC after CS, (c) Lanczos, and (d) JEC after Lanczos.

Table 5 PSNRs (dB) of various deinterlacing methods with and without the proposed JEC method

	CS		Lanczos		STCAD		MAVTF	
	Initial	JEC	Initial	JEC	Initial	JEC	Initial	JEC
BUS	28.44	<i>28.44</i>	28.43	<i>28.44</i>	28.24	<i>28.26</i>	28.78	<i>28.87</i>
Coastguard	29.08	<i>29.15</i>	29.31	<i>29.35</i>	31.38	<i>31.79</i>	31.23	<i>31.67</i>
Container	28.96	<i>29.16</i>	29.08	<i>29.26</i>	35.55	<i>35.80</i>	35.45	<i>35.73</i>
Foreman	34.31	<i>35.46</i>	34.50	<i>36.07</i>	36.43	<i>37.47</i>	36.78	<i>37.88</i>
Hall monitor	32.10	<i>32.22</i>	32.20	<i>32.34</i>	38.25	<i>38.34</i>	37.75	<i>38.31</i>
Mom daughter	33.38	<i>33.39</i>	33.67	<i>33.80</i>	38.14	<i>38.08</i>	38.30	<i>38.46</i>
Silent	34.13	<i>34.26</i>	34.38	<i>34.58</i>	39.93	<i>39.21</i>	40.33	<i>40.76</i>
Rodmap	28.49	<i>28.44</i>	28.09	<i>28.49</i>	30.25	<i>30.50</i>	29.85	<i>30.22</i>
Washdc	32.49	<i>32.39</i>	32.23	<i>32.24</i>	38.14	<i>38.37</i>	37.70	<i>37.43</i>
Mobile	32.54	<i>32.75</i>	32.84	<i>33.11</i>	34.53	<i>34.61</i>	34.90	<i>35.08</i>
Stockholm	32.55	<i>32.66</i>	32.88	<i>32.97</i>	33.83	<i>33.83</i>	33.07	<i>33.29</i>
Shields	30.99	<i>30.96</i>	31.25	<i>31.33</i>	30.92	<i>31.77</i>	30.70	<i>30.78</i>
Average	31.45	<i>31.60</i>	31.57	<i>31.83</i>	34.63	<i>34.84</i>	34.57	<i>34.87</i>

The JEC algorithm was applied to the deinterlaced image obtained by the initial deinterlacing methods. The higher PSNR values are set in italics.

MAVTF, and the proposed method, respectively. As shown in Figure 13, the existing methods produced jagged edge artifacts along the diagonal edges in red circled regions, but the proposed method provided satisfactory outputs without jagged edge artifacts. Tables 1 and 2 show the PSNRs and the MSSIMs of various deinterlacing methods, respectively. As described in Table 1, the proposed method recorded higher PSNR values than conventional deinterlacing methods in the majority of the test sequences. According to Table 2, the proposed method recorded higher MSSIM values than conventional deinterlacing methods in the majority of the test sequences. These experimental results demonstrate that the proposed method can be used for deinterlacing process to improve image quality.

In Figure 14, the performance of the proposed method was evaluated for the VDFC process. For this experiment, the test video sequences were downsampled by a factor of 2 and then these sequences were converted into an interlaced format. Again, the downsampled and interlaced sequences were deinterlaced with conventional deinterlacing methods and then progressive sequences were upsampled with conventional image upscaling methods. The STCAD was used to convert the interlaced format into a progressive format because the STCAD achieved high PSNR and SSIM values among the conventional deinterlacing methods (as shown in Tables 1 and 2). Figure 14a,b,c,d represents the results obtained from CS, Lanczos, NEDI, and SAI after performing STCAD. As shown in the red circled regions of Figure 14, LFI-based image upsampling methods in Figure 14a,b suffered from jagged edge artifacts along the diagonal edges, and EDI-based image upsampling methods in Figure 14c,d

provided fine results along the diagonal edges. However, NEDI and SAI introduced some artifacts in the neighborhood regions of the character regions due to an incorrect estimation of covariance. However, the proposed method produced high-quality images by connecting discontinuous edges and reduced the jagged edge artifacts substantially. Tables 3 and 4 present the PSNRs and the MSSIMs of various VDFC methods, respectively. As described in Table 3, the proposed method recorded higher PSNR values than conventional VDFC methods in

Table 6 PSNRs (dB) of various image upscaling methods with and without the proposed JEC method

	CS		Lanczos	
	Initial	JEC	Initial	JEC
Bus	25.85	<i>25.89</i>	26.21	<i>26.32</i>
Coastguard	28.42	<i>28.42</i>	28.87	<i>28.90</i>
Container	26.97	<i>27.06</i>	27.33	<i>27.49</i>
Foreman	33.60	<i>34.03</i>	34.14	<i>35.53</i>
Hall monitor	28.63	<i>28.68</i>	29.15	<i>28.28</i>
Mom daughter	32.22	<i>32.13</i>	32.45	<i>32.75</i>
Silent	<i>31.93</i>	31.90	32.40	<i>32.61</i>
Rodmap	<i>27.37</i>	27.01	<i>27.22</i>	<i>27.76</i>
Washdc	28.46	<i>28.46</i>	28.55	<i>28.71</i>
Mobile	29.86	<i>29.96</i>	30.31	<i>30.56</i>
Stockholm	31.07	<i>31.11</i>	31.57	<i>31.66</i>
Shields	29.15	<i>29.12</i>	29.80	<i>29.92</i>
Average	29.47	<i>29.48</i>	29.83	<i>30.12</i>

The JEC algorithm was applied to the enlarged image obtained by the initial image upscaling methods. The higher PSNR values are set in italics.

Table 7 SSIMs of various deinterlacing methods with and without the proposed JEC method

	CS		Lanczos		STCAD		MAVTF	
	Initial	JEC	Initial	JEC	Initial	JEC	Initial	JEC
Bus	0.912	<i>0.912</i>	0.913	<i>0.913</i>	0.910	<i>0.912</i>	0.915	<i>0.917</i>
Coastguard	<i>0.831</i>	0.828	<i>0.838</i>	0.833	0.918	<i>0.923</i>	0.916	<i>0.919</i>
Container	0.911	<i>0.913</i>	0.916	<i>0.917</i>	0.984	<i>0.985</i>	0.982	<i>0.984</i>
Foreman	0.940	<i>0.946</i>	0.944	<i>0.951</i>	0.958	<i>0.961</i>	0.960	<i>0.964</i>
Hall monitor	0.967	<i>0.968</i>	0.968	<i>0.969</i>	0.978	<i>0.978</i>	0.975	<i>0.978</i>
Mom daughter	0.946	<i>0.947</i>	0.949	<i>0.950</i>	0.969	0.967	0.972	0.971
Silent	0.927	<i>0.929</i>	0.931	<i>0.933</i>	0.972	0.970	0.987	<i>0.988</i>
Rodmap	0.938	<i>0.939</i>	0.936	<i>0.940</i>	0.962	<i>0.964</i>	0.964	<i>0.965</i>
Washdc	0.955	<i>0.955</i>	0.950	<i>0.955</i>	0.988	<i>0.989</i>	0.981	<i>0.983</i>
Mobile	0.908	<i>0.912</i>	0.916	<i>0.919</i>	0.932	<i>0.934</i>	0.936	<i>0.938</i>
Stockholm	0.890	<i>0.892</i>	0.898	<i>0.898</i>	0.913	0.911	0.915	<i>0.915</i>
Shields	0.913	<i>0.913</i>	0.918	<i>0.918</i>	0.921	<i>0.922</i>	0.916	<i>0.919</i>
Average	0.919	<i>0.921</i>	0.923	<i>0.924</i>	0.950	<i>0.951</i>	0.951	<i>0.953</i>

The JEC algorithm was applied to the deinterlaced image obtained by the initial deinterlacing methods. The higher SSIM values are set in italics.

the majority of the test sequences. According to Table 4, the proposed method recorded higher MSSIM values than conventional VDFC methods. From Tables 3 and 4, it can be verified that the proposed method outperformed the conventional approaches in terms of numerical values.

The performance of the proposed JEC method was also evaluated. We applied the proposed method to the deinterlaced images and the upscaled images. Figure 15a,c,e,g represents the deinterlaced results of CS, Lanczos, STCAD, and MAVTF, respectively. Using each figure as an input image, the proposed JEC method was performed to obtain the results in Figure 15b,d,f,h. Figure 16a,c represents the upscaled results of the CS and Lanczos, respectively. Using each figure as an input image, the proposed method obtained the results in Figure 16b,d. As shown in Figures 15 and 16, the jagged edge artifacts were reduced substantially by the proposed method. Tables 5 and 6 present the PSNRs of various deinterlacing methods and image upscaling methods, respectively. As described in Table 5, the proposed method improved the PSNR of CS, Lanczos, STCAD, and MAVTF by 0.15, 0.26, 0.21, and 0.3, respectively. As described in Tables 6, the proposed method improved the PSNR of CS and Lanczos by 0.01 and 0.29, respectively. In Tables 7 and 8, the MSSIMs of various deinterlacing methods and image upscaling methods are compared. As described in Table 7, the proposed method improved the MSSIM of CS, Lanczos, STCAD, and MAVTF by 0.002, 0.001, 0.001, and 0.002, respectively. According to Table 8, the proposed method improved the MSSIM of CS and Lanczos by 0.001 and 0.003, respectively. These experimental results demonstrate that the proposed JEC method can be used

as a postprocessor for various deinterlacing methods and image upscaling methods in order to improve the image quality.

To show the computational requirements, the average run times of various image formats were calculated (as shown in Table 9). For this experiment, we used a PC equipped with an Intel Core2 Quad Q8200 CPU. Especially, the resampled images of CS, Lanczos, NEDI, and SAI were obtained after performing the SAVTF. Thus, the processing time of SAVTF was added to

Table 8 SSIMs of various image upscaling methods with and without the proposed JEC method

	CS		Lanczos	
	Initial	JEC	Initial	JEC
Bus	0.85	<i>0.85</i>	0.856	<i>0.857</i>
Coastguard	<i>0.793</i>	0.792	0.802	<i>0.802</i>
Container	0.872	<i>0.873</i>	0.882	<i>0.884</i>
Foreman	0.923	<i>0.926</i>	0.930	<i>0.938</i>
Hall monitor	0.918	<i>0.919</i>	0.927	<i>0.929</i>
Mom daughter	0.930	<i>0.930</i>	0.933	<i>0.937</i>
Silent	0.887	<i>0.888</i>	0.892	<i>0.895</i>
Rodmap	<i>0.924</i>	0.921	0.922	<i>0.929</i>
Washdc	0.904	<i>0.905</i>	0.907	<i>0.911</i>
Mobile	0.842	<i>0.845</i>	0.854	<i>0.859</i>
Stockholm	0.844	<i>0.845</i>	0.856	<i>0.857</i>
Shields	0.878	<i>0.878</i>	0.885	<i>0.888</i>
Average	0.880	<i>0.881</i>	0.887	<i>0.890</i>

The JEC algorithm was applied to the enlarged image obtained by the initial image upscaling methods. The higher SSIM values are set in italics.

Table 9 CPU times of various VDFC methods including the proposed method for various image formats

	Image format	CS	Lanczos	NEDI	SAI	Proposed method
CPU	CIF (352 × 288)	0.049	0.013	0.898	4.344	0.833
time (s)	SD (720 × 486)	0.198	0.047	2.428	16.662	3.031
	HD (1,280 × 720)	0.443	0.115	8.498	39.433	7.578
	Full HD (1,920 × 1,080)	1.005	0.259	19.829	86.689	16.812

Resampled images of CS, Lanczos, NEDI, and SAI were obtained after performing the STCAD.

the total processing times of the conventional methods. As described in Table 9, the processing times increased depending on the resolution of the image format. Although EDI-based methods needed more time than LFI-based methods due to requiring many operations to estimate the edge direction, the EDI-based methods provided high-quality results and the LFI-based methods suffered from jagged edge artifacts in the diagonal edge regions. According to Table 9, the processing time of the proposed method was similar to that of NEDI. Thus, the proposed method and the NEDI method have similar complexity levels. However, the proposed method provided better objective and subjective performance than the other methods. Furthermore, the JEC process of the proposed method can be used as a post-processor for performance improvement of many linear filtering interpolation methods. Thus, either the total proposed method or the JEC method can be selectively used according to the applications.

5 Conclusions

In this paper, we have proposed a flexible video resampling method with the advantages of both the LFI and EDI methods: the capabilities of converting image formats, resizing images for arbitrary ratios and improving edge quality. The proposed method converted input interlaced sequences into upscaled progressive sequences simultaneously and improved the image quality of resampled images by correcting various interpolation artifacts, such as ringing, blurring, and jagged edge artifacts. In order to reduce the ringing artifacts, the proposed ringing reduction method was combined with Lanczos interpolation and spatio-temporal interpolation. Also, the proposed JEC method was applied to initially upscaled images to correct jagged edge artifacts and to improve the sharpness of the images. Especially, this JEC postprocessor can be very useful for various image resampling applications since it is often used in combination with other common LFI techniques. The proposed algorithm was applied to various test images to verify the performance. Simulation results show that the proposed method outperformed conventional methods both visually and numerically.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2012R1A2A4A01003732).

Received: 31 July 2013 Accepted: 2 December 2013

Published: 20 December 2013

References

- G de Haan, Large-video-display-format conversion. *J. Soc. Inf. Display* **8**(1), 79–87 (2000)
- G de Haan, EB Bellers, Deinterlacing - an overview. *Proc. IEEE* **86**(9), 1839–1857 (1998)
- GG Lee, H-Y Lin, M-J Wang, R-L Lai, CW Jhuo, B-H Chen, Spatial-temporal content-adaptive deinterlacing algorithm. *IET Image Process.* **2**(6), 323–336 (2008)
- K Lee, J Lee, C Lee, Deinterlacing with motion adaptive vertical temporal filtering. *IEEE Trans. Consum. Electron.* **55**(2), 636–643 (2009)
- Y-L Chang, S-F Lin, C-Y Chen, L-G Chen, Video deinterlacing by adaptive 4-field global/local motion compensated approach. *IEEE Trans. Circuits Syst. Video Technol.* **15**(12), 1569–1582 (2005)
- Y-R Chen, S-C Tai, True motion compensated deinterlacing algorithm. *IEEE Trans. Circuits Syst. Video Technol.* **19**(10), 1489–1498 (2009)
- Q Huang, D Zhao, S Ma, W Gao, H Sun, Deinterlacing using hierarchical motion analysis. *IEEE Trans. Circuits Syst. Video Technol.* **20**(5), 673–686 (2010)
- CE Shannon, Communication in the presence of noise. *Proc. I.R.E.* **M1-2**, 31–39 (1983)
- TM Lehmann, C Gönner, Survey: interpolation methods in medical image processing. *IEEE Trans. Med. Imaging* **18**(11), 1049–1075 (1999)
- RG Keys, Cubic convolution interpolation for digital image processing. *IEEE Trans. Acoustics, Speech Signal Process.* **29**(6), 1153–1160 (1981)
- HS Hou, HC Andrews, Cubic splines for image interpolation and digital filtering. *IEEE Trans. Acoustics, Speech Signal Process.* **508**(6), 517 (1978)
- T TheuBl, H Hauser, ME Gröller, Mastering windows: improving reconstruction, in *Proceedings of the 2000 IEEE Symposium on Volume Visualization* (Salt Lake City, 9–10 Oct 2000), pp. 101–108
- MK Park, MG Kang, K Nam, SG Oh, New edge dependent deinterlacing algorithm based on horizontal edge pattern. *IEEE Trans. Consum. Electron.* **49**(4), 1508–1512 (2003)
- S-J Park, J Jeong, Local surface model-based deinterlacing algorithm. *Opt. Eng.* **50**(1), 017004-1–017004-10 (2011)
- X Li, New edge-directed interpolation. *IEEE Trans. Image Process.* **10**(10), 1521–1527 (2001)
- WS Tam, CW Kok, WC Siu, Modified edge-directed interpolation for images. *J. Electron. Imaging* **19**(1), 013011 (2010)
- X Zhang, X Wu, Image interpolation by adaptive 2-D autoregressive modeling and soft-decision estimation. *IEEE Trans. Image Process.* **17**(6), 887–896 (2008)
- CH Park, J Chang, MG Kang, Kernel-based image upscaling method with shooting artifact reduction. *Proc. SPIE 8655 Image Processing: Algorithms and Systems XI* (2013). doi: 10.1117/12.2003326
- X Feng, P Milanfar, Multiscale principal components analysis for image local orientation estimation, in *Conference record of the 36th Asilomar Conference on Signals, Systems and Computers* (Pacific Grove, 3–6 Nov 2002), pp. 478–482
- J Chang, DS Yoo, JH Park, SH Park, MG Kang, Edge directional interpolation for image upscaling with temporarily interpolated pixels. *IET Electron. Lett.* **47**(21), 1176–1178 (2011)

21. H Takeda, S Frsiu, P Milanfar, Kernel regression for image processing and reconstruction. *IEEE Trans. Image Process.* **16**(2), 349–366 (2007)
22. P Lin, YT Kim, An adaptive color transient improvement algorithm. *IEEE Trans. Consum. Electron.* **49**(4), 1326–1329 (2003)
23. K Ohara, Digital color transient improvement. U.S. Patent, 5,920,357, 6 July 1999
24. J Chang, GS Shin, JH Park, MG Kang, Color transient improvement for signals with a bandlimited chrominance component. *Opt. Eng.* **46**(2), 027002.1–027002.11 (2007)
25. Z Wang, AC Bovik, HR Sheikh, EP Simoncelli, Image quality assessment: from error visibility structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004)

doi:10.1186/1687-6180-2013-188

Cite this article as: Yoo et al.: Video resampling algorithm for simultaneous deinterlacing and image upscaling with reduced jagged edge artifacts. *EURASIP Journal on Advances in Signal Processing* 2013 **2013**:188.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Immediate publication on acceptance
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

Submit your next manuscript at ▶ springeropen.com
