

# GPU-based real-time super-resolution system for high-quality UHD video up-conversion

Dae Yeol Lee<sup>1</sup>  · Jooyoung Lee<sup>1</sup> · Ji-Hoon Choi<sup>2</sup> · Jong-Ok Kim<sup>2</sup> · Hui Yong Kim<sup>1</sup> · Jin Soo Choi<sup>1</sup>

Published online: 13 September 2017  
© Springer Science+Business Media, LLC 2017

**Abstract** Super-resolution (SR) is a technique that reconstructs high-resolution images using the information present in low-resolution images. Due to their potentials of being used in wide range of image and video applications, various SR algorithms have been studied and proposed in the literature until recently. However, many of the algorithms provide insufficient perceptual quality, possess high computational complexity, or have high memory requirement, which make them hard to apply on consumer-level products. Therefore, in this paper we propose an effective super-resolution method that not only provides an excellent visual quality but also a high-speed performance suitable for video conversion applications. The proposed super-resolution adopts self-similarity framework, which reconstructs the high-frequency (HF) information of the high-resolution image by referring to the image pairs generated from self-similar regions. The method further enhances the perceptual sharpness of the video through region-adaptive HF enhancement algorithm and applies iterative back projection to maintain its consistency with the input image. The proposed method is suitable for parallel processing and therefore is able to provide its superb visual quality on a high conversion speed through GPU-based acceleration. The experimental results show that the proposed method has superior HF reconstruction performance compared to other state-of-the-art upscaling solutions and is able to generate videos that are visually as sharp as the original high-resolution videos. On a single PC with four GPUs, the proposed method can convert Full HD resolution video into UHD resolution with real-time conversion speed. Due to its fast and

---

✉ Dae Yeol Lee  
daelee711@etri.re.kr

<sup>1</sup> Electronics and Telecommunications Research Institute, Daejeon, Republic of Korea

<sup>2</sup> School of Electrical Engineering, Korea University, Seoul, Republic of Korea

high-quality conversion capability, the proposed method can be applied on various consumer products such as UHD TV, surveillance system, and mobile devices.

**Keywords** GPU · Real-time · Region-adaptive · Self-similarity · Super-resolution · UHD video

## 1 Introduction

Major TV manufacturers have driven the TV industry forward by announcing ultra-high-definition televisions (UHDTVs) globally. Having two times the resolution of the conventional Full HD (FHD) TVs in vertical and horizontal direction, the UHDTVs are expected to provide consumers with more realistic viewing experiences. However, the UHDTVs are yet to meet the mass-market appeal due to the lack of native UHD contents. Full-scale shift to a UHD contents production workflow will take some time considering the high costs and efforts involved in upgrading the current infrastructure into a UHD capable one. For such a reason, the method of converting the existing contents into high-quality UHD contents is getting attention as an alternative solution that could vitalize the UHD market.

In relation to the high-quality upscaling, various super-resolution techniques have been studied and proposed until recently. Early studies on super-resolution focused on mathematically modeling the inverse of the image acquisition process. Such reconstruction-based methods use regularization methods on multiple adjacent frames to effectively solve the ill-posed inverse problem [1–3]. However, these reconstruction-based methods require high computational complexity due to the multi-frame registration process and are prone to quality degradations. Another popular approach is the example-based method, which reconstructs the fine details of the high-resolution images by referring to the database consisting of example image pairs. One branch of the example-based method harnesses the power of machine learning and compactly represents the information in the extrinsic training datasets into a sparse dictionary or a neural network [4–12]. Especially, the deep learning-based methods [8, 12] proposed in recent years show considerable improvement in PSNR performance compared to other state-of-the-art methods. However, the performance of the machine learning-based methods relies on relevant training datasets, and the methods require many convolution operations and high amount of intermediate memory buffers, making them hard to apply on high-resolution video conversion applications. Another branch of example-based method, which is closely relevant to the proposed work, uses self-similar examples within a single image to reconstruct a high-resolution image. The self-similarity-based method was first proposed by Glasner et al. [13], where the self-similar example patches of the input image within the same scale and across different scales are used to reconstruct high-resolution image. Freedman et al. [14] further supports the self-similarity assumption by showing that relevant example patches are highly probable on same image with small scaling factor. The self-similarity-based super-resolution (SSSR) methods not only require lower memory resources, but also provide comparable visual quality to other external database or multi-frame-based approaches and thus has become widely studied subject in image resolution conversion area [13–17]. The details of the related recent studies are presented in Sect. 2.

In this paper, we propose a novel self-similarity-based super-resolution method, which reconstructs and enhances the HF signal of the image to the level of original high-resolution image, without requiring any extrinsic datasets. The proposed method first reconstructs the high-frequency (HF) signals of the high-resolution image by extracting information from the self-similar LF–HF pairs generated from the input image. Then, different from other conventional SSSR methods [13–17], the proposed method applies perceptually derived, region-adaptive enhancement factors on the reconstructed HF signals to improve the perceptual sharpness of the video while minimizing the visual degradation from boosting artifacts. The reconstructed high-resolution image then goes through iterative back projection to maintain its consistency with the input image. In consideration of the applications with mass amount of data, such as FHD-to-UHD video conversion applications, the proposed method is implemented to run on platform with multiple GPU cards using the OpenCL [18] framework. The experimental results show that the proposed method generates images or videos that are better or comparable quality to the state-of-the-art methods, while having significantly faster computation time. The proposed method accelerated with four GPU cards supports FHD-to-UHD video up-conversion speed of 60 fps. The proposed SSSR solution can be applied to a file-based UHD video conversion software, real-time video converter hardware for distribution and play out, surveillance software, or various other applications requiring fast and high-quality resolution upscale.

The remainder of the paper is organized as follows. Section 2 reviews the related studies on super-resolution proposed in recent years and presents the contributions of the work presented in this paper. Section 3 describes the details of the proposed SSSR method. Section 4 describes the GPU implementation and optimization techniques for achieving the real-time conversion speed. Section 5 demonstrates the experimental settings and results of the proposed SSSR in comparison with other state-of-the-art SR methods [4–9, 11, 12, 16, 19–21]. Finally, conclusions are drawn in Sect. 6.

## 2 Related works

In overall aspect, there are largely two directions in recent super-resolution studies. First direction is low complexity oriented, which gives more emphasis on the practical use of the algorithm on consumer-level products [16, 19, 21–23]. These algorithms consider the hardware or software implementation aspect of the algorithm, and aim to provide near real-time processing for converting high-resolution image or video inputs. The algorithms generally restrict the computational complexity to a certain level, and within the restriction, aim to maximize the visual quality of the output image or video. For instance, Yang et al. [22] proposed edge-guided interpolation method for display devices which uses local gradient feature to obtain dominant edge direction. Giachetti et al. [21] proposed a real-time artifact-free image upscaling method termed as iterative curve-based interpolation (ICBI), which first applies local interpolation along the direction where the second-order image derivative is lower and then applies the iteratively refinement to preserve the edge details while minimizing the artifacts. Kang et al. [23] proposed real-time super-resolution method for digital zooming application by using directionally adaptive truncated constrained least-squares (TCLS) filter for image interpolation. Jun et al. [16] proposed self-example-based super-resolution

method and implemented the method on GPU-based platform for video upscaling application. Infognition presented a HD-to-UHD video up-conversion solution [19] that uses motion-search-based super-resolution, which extracts texture and edge detail information from adjacent frames. The aforementioned low-complexity oriented methods may not provide highest possible objective or subjective quality, but are able to provide reasonable quality output video on practical conversion speed.

Second direction is high quality oriented, where the quality of the output video is prioritized over the implementation suitability [4–12]. Though not necessarily slow, these algorithms consider less on the memory requirement or computational complexity and tend to employ machine learning techniques such as sparse representation or convolutional neural network (CNN) for enhancing the quality. For instance, Yang et al. [4], Peleg et al. [5], and Timofte et al. [7] proposed a super-resolution method based on the sparse representation of low- and high-resolution image patches. The methods construct sparse dictionary from the training LR–HR patch pairs and use it to reconstruct high-resolution images. Zeyde et al. [6] proposed various optimization methods with an aim to reduce the complexity of the traditional sparse representation based super-resolution. Dong et al. [10, 11] employ sparse representation framework on image de-noising, de-blurring and further on image super-resolution application as well. Especially in [11], Dong et al. propose nonlocally centralized sparse representation (NCSR) model which iteratively reduces the sparse coding noise (SCN) for image restoration. Choi et al. [9] proposed super-interpolation (SI) method that involves offline training phase and online upscaling phase, where the linear mapping function for various edge-orientation (EO) classes is obtained in the training phase. Because the SI method converts the input image directly to the target resolution without any intermediate interpolation, it requires relatively low hardware resources. Dong et al. [8] proposed SRCNN algorithm which employs three-layer CNN for LR–HR mapping. Kim et al. [12] proposed VDSR algorithm which uses deep CNN of twenty layers for residual learning. Though the deep learning-based methods such as SRCNN and VDSR have shown considerable improvement in PSNR performance compared to other state-of-the-art methods, they are computationally complex due to the 64 convolution filter operations on each intermediate layers and also require high amount of memory for holding the intermediate results. Thus, they are not suitable for applying on low-resource hardware as of yet. The aforementioned quality-oriented methods aim to push the limit of possible PSNR and SSIM values that are achievable through super-resolution methods, and the performance of the algorithms is often demonstrated through the PSNR and SSIM comparisons on standard test images.

Between the two directions, the proposed method positions itself on a low-complexity-oriented SR method. The proposed method aims to provide competitive visual quality to the state-of-the-art methods, while having much faster video conversion speed through GPU-based acceleration. The main contributions of our work are as follows. First, we exploit how HF signal adjustments on different image regions affect the perceptual quality of the image and employ the relationship to the SSSR framework to maximize the perceptual sharpness of the image. Second, we achieve the aforementioned improvements on perceptual sharpness and HF reconstruction performance while having minimal perceptually noticeable artifacts, through applying perceptually derived region-adaptive factors on HF signals, and optimizing the

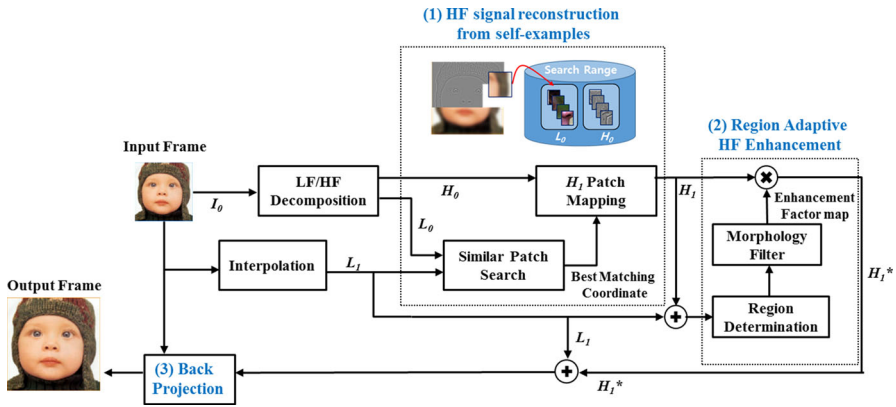


Fig. 1 Block diagram of the proposed SSSR method

interpolation kernel. Third, we prove that the proposed method presents competitive quality to the aforementioned state-of-the-art methods while having a significantly faster processing time. The performance comparison results of the proposed method with respect to the state-of-the-art methods are provided in detail in Sect. 5.

### 3 Proposed SSSR method

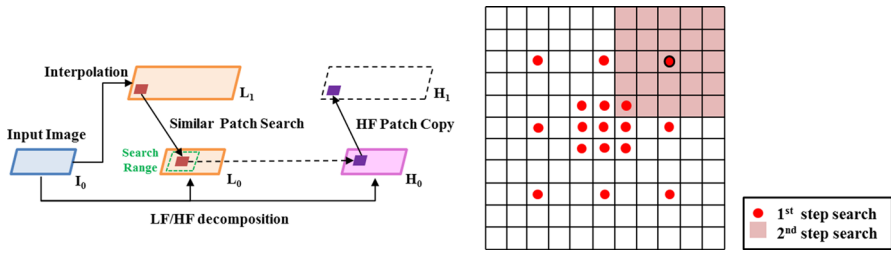
We present a novel self-example-based super-resolution method that effectively reconstructs the HF details of the high-resolution images from a single image. The proposed method consists of three procedures: (1) Reconstruct HF signal of the high-resolution image from self-examples, (2) region adaptively enhance the reconstructed HF signal to improve the perceptual sharpness of the image while minimizing visual artifacts, and (3) apply iterative back projection to maintain consistency with the input image. In this section, we provide details of each procedures and explain the interpolation kernel modification applied for improving the visual quality. Figure 1 shows the overall block diagram of the proposed SSSR method.

#### 3.1 HF signal reconstruction from self-examples

Natural images tend to contain lots of repeating patterns within and across different scales. This self-similar property of the image has served as a basis for many image-processing algorithms such as de-noising and texture synthesis. The proposed method also exploits the self-similar redundancy of the natural image, extracts relevant LF–HF example pairs from the input image itself, and uses them to reconstruct the HF signal of the high-resolution image. Figure 2 left shows the HF signal reconstruction procedure. The input image, denoted as  $I_0$ , is first decomposed into LF component  $L_0$  and HF component  $H_0$  using the equations below.

$$L_0 = I_0 * G \tag{1}$$

$$H_0 = I_0 - L_0 \tag{2}$$



**Fig. 2** The HF signal reconstruction process using self-examples (*left*) and the proposed CTSS search pattern (*right*)

The  $L_0$  is generated by convolving a low-pass operator, Gaussian blur kernel  $G$ , on  $I_0$ . The corresponding  $H_0$  is generated by subtracting  $L_0$  from the input image  $I_0$ . The image patches extracted from  $L_0$  and  $H_0$  will serve as a LF–HF pair database.

In the next step, the input image  $I_0$  is interpolated to a higher-resolution image  $L_1$ . The interpolation method used here is a modified version of the bicubic interpolation, and the details will be provided in later section. Previous studies experimented on how local self-similarity holds on various scaling factor, and the results show that low scaling factors are desirable for exploiting the self-similar information [14, 15]. Therefore, instead of reaching the target resolution at once, many self-similarity-based super-resolution methods upscale the input image gradually by a small scale factor (e.g., 1.25) multiple times. The proposed method also employs this scheme and upscales the input image by factor of 1.25 iteratively until the target resolution is reached. The upscaled image  $L_1$  is considered as the LF component of the upper resolution image, since it lost some amount of HF components during the interpolation process. The corresponding upper resolution HF plane  $H_1$  is restored through a similar patch search process.

The similar patch search process is conducted between the interpolated image  $L_1$  and  $L_0$  on a patch basis, consisting of  $5 \times 5$  pixels. The interpolated image  $L_1$  is regarded as a group of LF patches of the upper resolution image, and the input image  $I_0$ , possessing both LF and HF components, is regarded as a database containing LF–HF patch pairs. The goal here is to predict the unknown HF component of the upper resolution image  $H_1$  by referring to the database containing the relationship between LF and HF image patches. For each image patch of the  $L_1$ , the most similar patch is searched and selected from  $L_0$  and the corresponding patch from is  $H_0$  mapped to the upper resolution’s HF component. The interpolated image can be denoted as  $L_1 \in \mathbb{R}^{sR_1 \times sC_1}$ , where  $s$  refers to the scaling factor and  $R_1$  and  $C_1$  refers to the row and column resolution of the input image, respectively. For each query image patches of  $L_1 \in \mathbb{R}^{R_1 \times C_1}$ . The best matching patch is determined to be the patch with the lowest cost function, which is calculated by the sum of absolute difference (SAD) as below.

$$SAD(x, y, u, v) = \sum_{i=-\lfloor \frac{p_1}{2} \rfloor}^{\lfloor \frac{p_1}{2} \rfloor} \sum_{j=-\lfloor \frac{p_2}{2} \rfloor}^{\lfloor \frac{p_2}{2} \rfloor} |L_1(x+i, y+j) - L_0(u+i, v+j)| \tag{3}$$

The  $(p_1, p_2)$  refers to the image patch dimensions,  $(x, y)$  refers to center coordinate of the query patch, and  $(u, v)$  refers to the center coordinates of the search candidate patches in  $L_0$ . The search candidate patches are chosen from a local search area (e.g.,  $11 \times 11$  search window) in  $L_0$  centered around the relative coordinate of the query patch  $(\lfloor \frac{x}{s} \rfloor, \lfloor \frac{y}{s} \rfloor)$ , since it is proven that relevant patches can be best found in the restricted relative neighborhoods when upscaling with a small factor [14, 15]. Also, instead of calculating SADs on every point in the search area, a hierarchical search from coarser to finer grid was considered as an effective way to reduce the search points. According to the similar patch search experiment conducted by Yang et al. [15], the patches with lower matching errors are highly localized around the in-place position of query patch on a small scaling factor. Therefore, as shown in Fig. 2 right, we applied a center-biased hierarchical search pattern called center-biased two-step search (CTSS) to reduce the computational complexity while minimally affecting the visual quality. The procedures of the CTSS are as follows:

- In the first step, the algorithm starts from the center of  $11 \times 11$  search window. Set the initial step size  $S = 3$  to conduct patch search on a coarse grid. The center pixel along with the eight pixels at the location of  $+/- S$  is chosen as initial search points.
- To reflect the center-biased nature of the similar patch search in between the scales, set the step size  $S = 1$  and add additional eight pixels at the location of  $+/- S$  from the center to the initial search points.
- Calculate the SAD values between the query patch and the patches centered on the initial search points. Find the point with the least SAD value.
- In the second step, to conduct a fine search on a search area of  $5 \times 5$  centered around the point selected from the first step. The point with the least SAD value becomes the center point of the best matching patch.

For search area of  $11 \times 11$ , the full search requires 121 SAD calculations, while CTSS requires only 42 SAD calculations. The CTSS approximately reduces the number of calculation by one-thirds while having negligible effect on visual quality. Once the best matching patch is selected from  $L_0$ , the corresponding patch in  $H_0$  plane is mapped to the  $H_1$  plane to be considered as a HF component of  $L_1$  query patch.

In order to preserve the inter-patch relationship between the adjacent patches, we sample the  $L_1$  query patches to have overlapping area to each other. The stride level of the  $L_1$  query patch is set to 3 in vertical and horizontal directions, and therefore, each patches have 2 pixels overlapped with its neighbor patch on all boundaries. The reconstructed HF patches also have the corresponding overlap areas. The HF patches are aggregated into a  $H_1$  plane using the equations below.

$$w_i = \exp\left(-\frac{\widehat{SAD}_i}{\sigma^2}\right) \tag{4}$$

$$H_1(x, y) = \frac{\sum_{k \in N(x,y)} w_k \widehat{H}_k(x, y)}{\sum w_k} \tag{5}$$



The Eq. (4) shows weighting function  $w_i$  that considers the self-similarity level of the  $i$ th  $L_1$  query patch, and  $\widehat{\text{SAD}}_i$  refers to the sum of absolute difference between the  $i$ th  $L_1$  query patch and the corresponding best matching patch from  $L_0$ . The  $\sigma^2$  controls the degree of similarity, which is empirically set to 60. The Eq. (5) shows how HF patches with overlap areas are aggregated into a  $H_1$  plane. The  $(x, y)$  in Eq. (5) refers to the spatial coordinate in  $H_1$  plane.  $N_{(x,y)}$  refers to the set of patch indexes that contains coordinate  $(x, y)$ , and there may exist multiple elements in  $N_{(x,y)}$  if  $(x, y)$  is in the overlap area. The  $\widehat{H}_k$  refers to the reconstructed HF patch of the  $k$ th query patch. Considering that some spatial regions in  $H_1$  may have multiple reconstructed HF patches overlapped, Eq. (5) merges multiple HF patches on such region using the weighted average operation, where the weights are derived from (4). By applying such neighbor embedding method, the HF information reconstructed on a patch basis is merged into a plane while preserving the local relationship between patches. The reconstructed  $H_1$  plane is passed onto the region-adaptive HF enhancement procedure for increasing perceptual sharpness.

### 3.2 Region-adaptive high-frequency enhancement

Various SSSR methods [13–17] have reconstructed the HF signal from self-examples and presented a visually sharper image compared to the simply interpolated image. However, the amount of the HF signals reconstructed through such methods is still insufficient when compared to the HF signals of the original high-resolution video. A simple solution of multiplying a magnification factor on the reconstructed HF signals has been deployed by Park et al. [24]. However, such uniform HF amplification introduces visually noticeable artifacts especially in flat regions of the video, thereby severely degrading the perceptual quality. To overcome the problem, we propose a new region-adaptive HF enhancement method that effectively increases the visual sharpness while minimizing the visual degradation from artifacts.

As demonstrated from various literature works, the noticeability of the artifacts is highly correlated with the texture complexity of the interest region [25–27]. In order to derive the artifact noticeability threshold for our SSSR application scenario on different image regions, we conducted a subjective experiment on videos and still images. For the experiment, we collected videos and still images from the TID 2013 image database [28] and SJTU 4K video database [29]. The videos and still images were deliberately down-sampled by half resolution in horizontal and vertical direction and then upsampled back to the original resolution through SSSR method. The HF enhancement factor for the HF signal was controllable by the subjective experiment coordinator. Twenty participants with normal vision were guided to focus on different video or image areas having various texture complexity. The experiment coordinator sequentially increased the HF enhancement factor by 0.1 until the participant indicated that the artifact was noticeable, and the enhancement factor was recorded down as the artifact noticeability threshold of the area. The collected subjective experiment results consist of the area's texture complexity [30] and its corresponding HF enhancement factor in which the artifact was noticeable. The areas are then classified into flat, edge, or texture region by thresholding the difference curvature map as in [30]. Table 1



**Table 1** Average HF enhancement factor threshold for noticeable artifacts

	Flat	Edge	Texture
Still image	1.8	2.0	2.6
Video	1.5	2.0	1.5

shows the average HF enhancement factor for each regions where the artifacts became noticeable.

As shown in Table 1, the result turned out differently for still images and videos. For the still image’s case, the result well reflects the texture masking effect, where the visibility of the target artifact decreases on maskers having complex texture or similar frequency with the target [25]. However, in the video’s case, a severe flickering effect was observed when increasing the enhancement factor of the texture region. Therefore, the threshold for the noticeable artifact in texture region was presented relatively low when compared to the still image’s case. Unlike the relatively stable flat and edge regions, the texture region possesses a complex structure that is variant on different frames. Therefore, there are higher chance of having differently shaped HF signals on colocated texture region of the adjacent frames. When such HF signals are enhanced by a high factor, this translates into a visible flickering artifact, which was not observable in still images.

As the results in Table 1 illustrate, images and videos on different regions have different artifact noticeability. Therefore, in order to minimize the visual degradations from the artifacts while enhancing the HF signals, the enhancement factors should be applied adaptively for each regions accordingly. Based on the observations, we first constructed a map that classifies different regions of the image and then applied region-adaptive enhancement factors derived from the perceptual experiment. For identifying the flat, edge, and texture regions, we adopted the aforementioned region classifier, difference curvature proposed by Chen et al. [30], formulated as below:

$$u_{\epsilon\epsilon} = \frac{u_y^2 u_{xx} - 2u_x u_y u_{xy} + u_x^2 u_{yy}}{u_x^2 + u_y^2} \tag{6}$$

$$u_{\eta\eta} = \frac{u_x^2 u_{xx} + 2u_x u_y u_{xy} + u_y^2 u_{yy}}{u_x^2 + u_y^2} \tag{7}$$

where  $u_x$  and  $u_y$  refer to the first derivative gradient in  $x$  and  $y$  directions, respectively. The  $u_{xx}$ ,  $u_{yy}$ , and  $u_{xy}$  refer to the second derivative gradient in  $x$ ,  $y$ ,  $xy$  directions, respectively. The region where  $|u_{\eta\eta}|$  value is high and  $|u_{\epsilon\epsilon}|$  value is low is classified as edge region. The region where both  $|u_{\eta\eta}|$  and  $|u_{\epsilon\epsilon}|$  values are large is classified as a texture region. The region where both  $|u_{\eta\eta}|$  and  $|u_{\epsilon\epsilon}|$  values are small is classified as a flat region. Using the difference curvature, we first determine whether each pixel is flat, edge, or texture region. Then, to suppress the possible temporal flickering artifact from the pixel-wise region classification, we removed isolated bumps or holes of the region map using morphology filters consisting of erosion and dilation operations. Based on this morphology filtered region map, we constructed the enhancement factor

map by inputting the perceptually derived enhancement factors on each regions. The enhancement factor map is multiplied to the reconstructed  $H_1$  plane from previous procedure, and the enhanced HF plane is now termed as  $H_1^*$  plane.

### 3.3 Back projection

The  $H_1^*$  plane is combined with  $L_1$  to form an upper resolution image; then, the combined image goes through back projection operation to verify its consistency with the input image [13, 17]. The back projection is an algorithm intended to minimize the reconstruction error through iteratively compensating the difference between the simulated data and the observed data. In case of SSSR, the reconstruction error and the iterative compensation can be formulated as below:

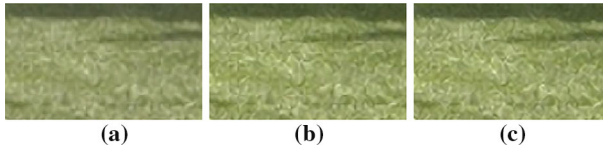
$$e(I) = I_0 - (h_{lp} * I) \downarrow_s \quad (8)$$

$$I_n^{k+1} = I_n^k + h_{bp} * e(I_n^k) \uparrow_s \quad (9)$$

where  $I_0$  refers to the original input image and  $I_n^k$  refers to the reconstructed image on  $n$ th resolution layer on  $k$ th back projection iteration. The  $h_{lp}$  and  $h_{bp}$  refer to the low-pass and back projection kernels, respectively, and  $*$  is the convolution operator. The  $\downarrow_s$  and  $\uparrow_s$  are the down-sampling and up-sampling operator with scaling factor  $s$ , which is set to 1.25 in our method. The upscaling method used here is a modified version of the bicubic interpolation, and the details will be provided in the next section. Because the reconstructed image may contain unnatural HF details that were not present in the source image, it is down-sampled and compared with the original input image. The resulting residuals from the comparison are upscaled back and applied to the reconstructed image to compensate the difference. After the back projection operation, the result image becomes the input image for the next resolution layer of the image pyramid, and the process is applied repeatedly by a small scaling factor until the designated resolution is reached.

### 3.4 Cubic spline tension value adjustment

The proposed SSSR method is able to generate a perceptually sharp images and videos, through reconstructing and enhancing the high-frequency signals on different image regions, notably on edge regions. However, we have observed that when using the proposed method, some misinterpreted edges pixels can be amplified as vertical and horizontal line artifacts. The line artifacts are especially observable when watching the video with a big display from a close distance. Through experiments, we have found that the choice of interpolation method affects these line artifacts. In this section, we show the effect of various interpolation methods on the output image and further suggest an optimal interpolation method for the proposed method. The suggested interpolation kernel possesses same computational complexity as the bicubic interpolation while reducing the line artifacts significantly.



**Fig. 3** Proposed method applied with different interpolation methods: **a** bilinear interpolation, **b** bicubic interpolation and **c** Lanczos interpolation

**Table 2** Properties of the interpolation methods

	Bilinear	Bicubic	Lanczos
Advantages	Low complexity	Moderate complexity Preserved details	Less line artifacts Preserved details
Disadvantages	Heavily blurred Line artifacts present	Line artifacts present	High complexity High memory requirements

Various single-image-based SR methods, including the proposed method, encompasses an interpolation operation that is used to enlarge the size of the input image or to rescale the images in the back projection process. Because the results of these interpolation operations are referred in the image reconstruction process, the quality of the output image is affected by the choice of interpolations. Based on the observation, we first conducted an experiment on the proposed method by varying the interpolation methods and investigating the output image. The interpolation methods include bilinear, bicubic, and Lanczos. Figure 3 shows the visual results of the experiment, and Table 2 shows a summary of the advantages and disadvantages of each interpolations on the proposed method.

As presented in Fig. 3, we observe that bicubic and Lanczos interpolation generate images with preserved details compared to heavily blurred bilinear interpolation result. However, in bicubic interpolation case, we see vertical and horizontal line artifacts that are visible when zoomed in or observed from a close distance. The line artifacts are also observable in bilinear interpolated image, but in a less apparent manner. Such line artifacts first appear when interpolating the image with a small scale factor (e.g., 1.25) as a segment of impulse pixels, and when such segment is misinterpreted as ‘edge’ pixels, they are amplified into visible line artifacts through series of procedures. Firstly, in the similar patch search process, the interpolated patch that contains the ‘false edge’ pixels is matched with the LF–HF pair of the input image having an edge-like structure on the corresponding location, thereby enhancing some amount of HF signals on the false edge. Secondly, in the region-adaptive HF enhancement step, the false edge pixels are classified as edge region and their HF signals are further enhanced accordingly. Thirdly, considering that the result image of the first resolution layer becomes the input image for the next resolution layer, the false edge pixels from first resolution layer will be considered as ground-truth edge pixels from the second resolution layer. Therefore, the HF signals of the false edge pixels are further amplified throughout multiple resolution layers, resulting in visible line artifacts.

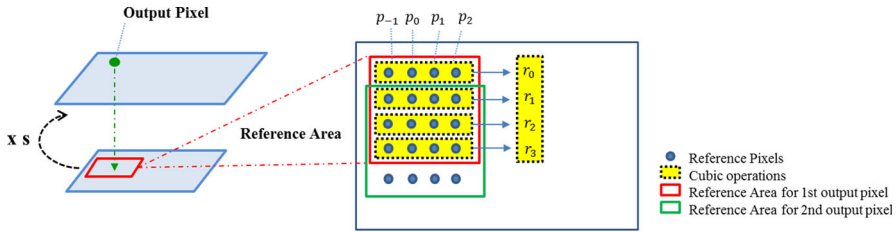


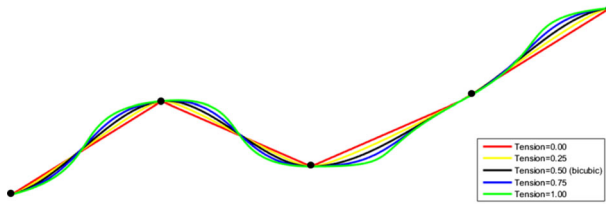
Fig. 4 Bicubic interpolation unrolled into five cubic operations

As shown in Fig. 3c, one solution for reducing the line artifacts is applying Lanczos interpolation, which provides images with reduced line artifacts while having the details preserved. However, in the implementation aspect, the Lanczos interpolation not only requires high memories for varying size kernels that are dependent on various resolutions of the proposed method’s image pyramid, but also requires high computations from sine-based kernel coefficient calculations and subsequent convolution operations. To overcome the problem, we present a simple but effective solution based on bicubic interpolation, which reduces the line artifacts while having low computational complexity. The bicubic interpolation considers sixteen reference points for reconstructing an output pixel value. As shown in Fig. 4, this bicubic interpolation can be unrolled into five cubic operations, which each refer to four reference values.

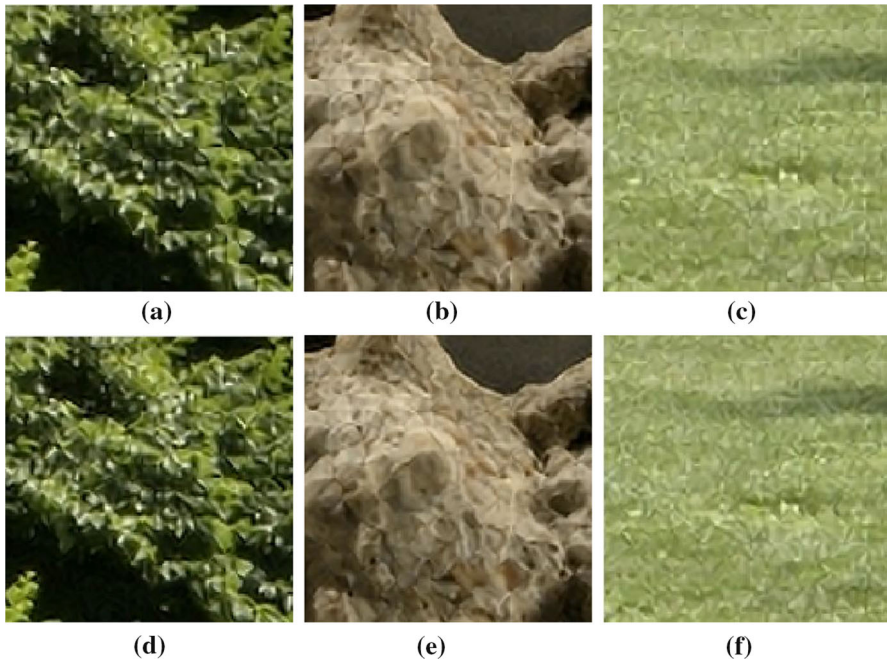
Four cubic operations are conducted on each lines of four pixels, and the fifth cubic operation is conducted on the calculated four result values. The cubic operation is formulated as below:

$$p(s) = \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\tau & 0 & \tau & 0 \\ 2\tau & \tau - 3 & 3 - 2\tau & -\tau \\ -\tau & 2 - \tau & \tau - 2 & \tau \end{bmatrix} \begin{bmatrix} p_{i-1} \\ p_i \\ p_{i+1} \\ p_{i+2} \end{bmatrix} \tag{10}$$

where  $p(s)$  refers to the output pixel value,  $u$  refers to the distance between the output pixel coordinate and the collocated coordinate in the reference area,  $p_i$  refers to the  $i$ th reference pixel value and  $\tau$  refers to the tension parameter having a value between 0 and 1. The basis matrix above follows the cardinal cubic spline definition, which indicates a series of curves joint together to form a larger continuous curve. The shape of the cardinal spline curve is determined by the set of control points and the tension parameter. As shown in Fig. 5, lower tension parameter corresponds to a higher physical tension, which results in a tighter curve connecting the control points. An extreme case is shown in tension parameter 0, where the infinite physical tension has forced the curve to take the shortest path in between the control points, resulting in a very impulsive curve. On the contrary, higher tension parameter corresponds to a curve with less physical tension, resulting in a looser curve. The curve with tension value 0.5 generates the same result as the bicubic interpolation. Considering that the proposed method induces line artifacts by miss interpreting and amplifying the impulsive segments from the bicubic interpolation, we can effectively reduce the artifacts by generating interpolation results that are less impulsive by applying tension parameter

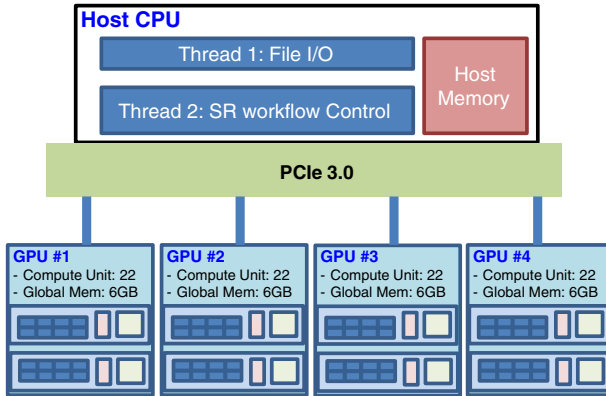


**Fig. 5** Cardinal cubic splines on different tension values



**Fig. 6** Proposed method applied with different interpolation methods: **a–c** for original bicubic interpolation ( $\tau = 0.5$ ), **d–f** for tension value adjusted bicubic interpolation ( $\tau = 1.0$ )

value higher than 0.5. Through various experiments, we have validated that the tension parameter of 1.0 best preserves good amount of visual details while effectively reducing the line artifacts. Figure 6 shows the visual results of applying proposed method using bicubic interpolation and the tension value adjusted interpolation. In the figure, we see that the line artifacts are significantly reduced by using the new interpolation. The new interpolation is also favorable in the implementation aspect that it maintains the complexity of the bicubic interpolation through only modifying the coefficients of the basis matrix. We have adopted this new interpolation in our implementation, and any interpolation operations used within our proposed SSSR method will refer to it.



**Fig. 7** The structure of the implemented system

## 4 Optimizations on GPU implementation

This section covers the details of the GPU implementation and optimization techniques applied on the proposed SSSR method. The proposed SSSR method was implemented using OpenCL framework to manage and utilize the resources in multiple GPU devices. As shown in Fig. 7, the implemented system consists of a host CPU and four GPUs connected with PCIe 3.0 link. The host CPU in our system possesses 40 lanes and is connected to four GPU cards by  $\times 8/\times 8/\times 8/\times 16$  configuration, respectively. Considering that the throughput of PCIe 3.0 is 985 MB/s for each lane and a single UHD YUV 4:4:4 10-bit raw frame has a data size of 47.4 MB/s, the maximum *I/O* throughput by a single  $\times 8$  connection is approximately 166 UHD frames per second. This *I/O* throughput is sufficient for our real-time system consisted of four GPU connections. The host CPU runs two main threads where one controls the file *I/O* of the system, and the other controls the SSSR workflow by sending execution commands for each step of the SSSR to multiple GPUs. The file *I/O* thread reads multiple frames from the input HD video file and delivers the frames to GPU cards for SSSR execution. After the host CPU receives back the result UHD frames from GPU cards, the file *I/O* thread writes the result UHD frames to the output file. After the file write is finished, next set of input frames are read for the next turn of the execution. The workflow thread controls four sub-threads, which sends out execution commands to four GPU cards consecutively via OpenMP API. After the GPU card receives the execution commands from host CPU, appropriate kernels for each steps are executed parallel on the many cores on GPU devices. In the following section, we introduce optimization techniques applied on kernel, memory model, and context management, which accelerated conversion speed of the system significantly.

### 4.1 Kernel optimization

The kernel in OpenCL refers to computation instances that are executed on the cores of the GPU. According to the execution model of the OpenCL platform, the execution

**Table 3** Computation time of major kernels

Kernels	Original (ms/frame)	Optimized (ms/frame)	Optimization description
Similar patch search	38.76	22.08	CTSS search pattern, speedup ratio of 1.76
Convolution	35.13	5.74	Separable kernel, speedup ratio of 6.12
Morphology	20.76	4.85	Separable kernel, speedup ratio of 4.28
Bicubic interpolation	8.32	–	–
Region determination	1.53	–	–
Addition	1.31	–	–
Subtraction	0.78	–	–
Multiplication	0.87	–	–
Division	1.02	–	–

of a kernel corresponds to a work item, which is handled by a processing element in GPU. A group of work item, termed as work group, can be executed in parallel by a compute unit. The dimension of a work group should be designed in consideration of the GPU memory resource and the resolution of the image to process. The kernels for the major functions in the proposed SSSR were designed as listed in Table 3.

Most of the kernels are straightforward in the implementation aspect, but some of the kernels possess high computational complexity and therefore need optimization technique to reduce the computation time. Among the kernels, the most time-consuming ones are similar patch search, convolution, and morphology kernel which each takes 35.7, 32.4, and 19.13% of the total computation time, respectively. The convolution kernel and morphology kernel can be accelerated losslessly using the separable property of the 2D filters. The convolution is used when decomposing the input image into LF and HF domain and is also used when applying anti-aliasing filter in the back projection process. The morphology kernel refers to erosion and dilation operations used in the region classification process to reduce the possible temporal flickering. Both kernels apply 2D filters on each pixels of the image and therefore possess high computational complexity. However, considering that the convolution operation has an associative property, the number of computations can be reduced by determining the separability of the 2D filters and decomposing them. Erosion and dilation operations are also separable if applying symmetrical structuring elements. The separability of the 2D filters is determined and decomposed using singular value decomposition (SVD). The kernel  $M$  can be expressed in the form of  $M = U \cdot \Sigma \cdot V'$ , where  $U$  and  $V$  refers to unitary matrix and  $\Sigma$  refers to a diagonal matrix that contains nonnegative singular values of matrix  $M$ . The decomposed vectors  $\mathbf{v1}$  and  $\mathbf{v2}$  can be derived using following equations:

$$\mathbf{v1} = U_1 \times \sqrt{\Sigma_{11}} \quad (11)$$

$$\mathbf{v2} = V_1' \times \sqrt{\Sigma_{11}} \quad (12)$$



where  $U_1$  and  $V_1$  refer to the first columns of  $U$  and  $V$ , respectively, and  $\Sigma_{11}$  refers to the nonzero singular value of the separable matrix. By applying the computations with the decomposed vectors, the computational complexity of the operations drops from  $O(L^2)$  to  $O(L)$  where  $L \times L$  refers to the dimension of  $M$ . The similar patch search kernel is also one of the kernel that possesses high computational complexity due to the SAD calculations for every points in the search area. Therefore, as explained previously, we applied a hierarchical CTSS search pattern, which effectively reduces the search points to one-thirds while having negligible effect on outputs. The speedup ratio of applying the aforementioned optimizations is presented in Table 3.

## 4.2 Memory optimization

The OpenCL platform defines a hierarchical memory model as shown in Fig. 8. The host memory from CPU transfers data to the GPU devices via PCIe 3.0 bus, and the transferred data are stored on global, local, or private memories in the device.

Considering that the data transfer using the PCIe 3.0 bus has much lower bandwidth compared to the global memory access, the data transfer between the host and the device is minimized by allocating all the intermediate data on the global memory of the GPU. The data transfer between the host memory and the global memory occurs only when sending the input frame and receiving the output frame.

While the global memory can be accessed by all work items within the device, the access to the local memory is limited to work items within the same work group. However, the access to local memory is much faster than the global memory. Thus,

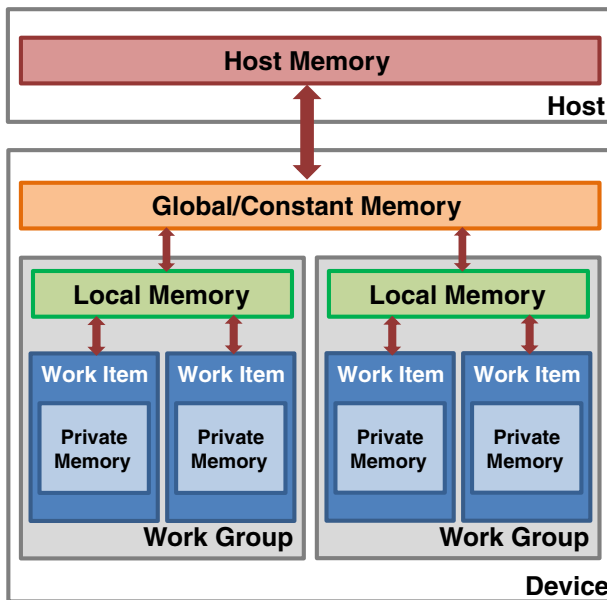
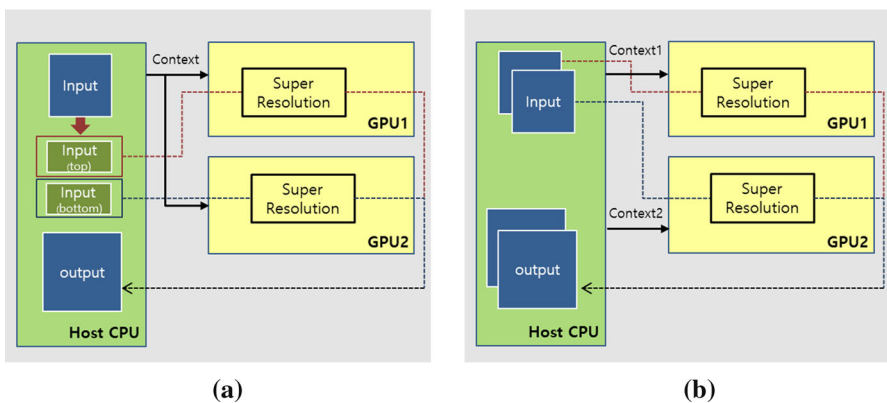


Fig. 8 Hierarchical memory model of OpenCL

if the kernel refers to the same image region repeatedly, it is beneficial to cache the region to the local memory. In our implementation, the similar patch search kernel refers to a search area of  $11 \times 11$  centered around the query patch, and consequently, there is a high chance of overlapped search area on when conducting similar patch search on nearby query patches. Therefore, the combined search area of the work group consisting of adjacent query patches is loaded all together to the local memory for optimal performance.

### 4.3 Context management

A proper context management is important for utilization of the resources in multiple GPUs. As shown in Fig. 9, a single context can be shared among multiple GPU devices, or multiple contexts can be created per devices. In the case of single context-based implementation, the OpenCL objects such as kernels, programs, and memory objects are shared among multiple GPU devices. When implementing the proposed SSSR on a single context, a single frame is split equally into multiple segments and is processed on each GPU using the shared kernels. Such method may require sending extra pixel data around the borderlines between the image segments on each GPU for precise calculations and may also require extra stitching operations on the resulting image, thereby increasing the total computation time. Multiple contexts-based implementation, on the other hand, creates redundant OpenCL objects for each GPU devices, thereby requiring more initial memories. However, the multiple context-based implementation adheres closely to the distributed programming and may be more intuitive in splitting the works to multiple devices. For instance, multiple contexts-based implementation of the proposed method distributes each frames to each GPU cards. Considering that the single context approach shows longer computation time compared to the multiple contexts approach, due to the data transfer overhead incurred from extra border line pixels [16], we adopted the multiple contexts management for our implementation.



**Fig. 9** Structures of context managements: **a** single context and **b** multiple contexts

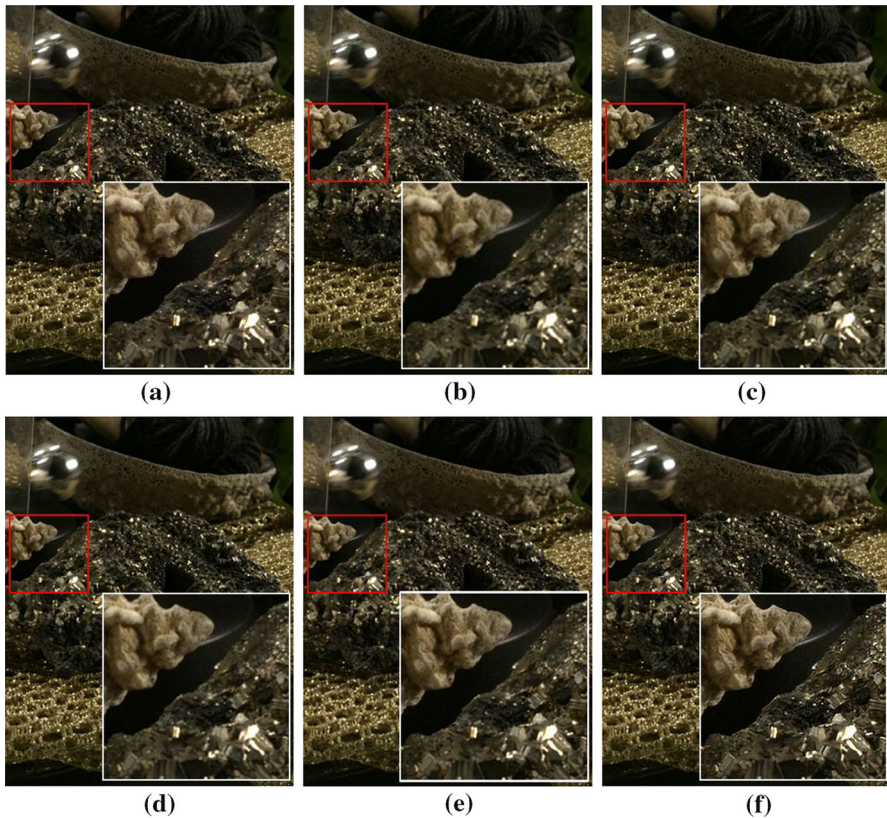
**Table 4** Description of UHD test sequences

Sequence	Screenshot	Description of the sequence
Park Dancer [31]		<p>Motion level: high</p> <ul style="list-style-type: none"> <li>-Contains complex motions from the dance movements of the dancers</li> </ul> <p>Spatial complexity: high</p> <ul style="list-style-type: none"> <li>-Contains complex textures on grasses and trees in the background</li> </ul>
Pendulus Wide [31]		<p>Motion level: medium</p> <ul style="list-style-type: none"> <li>-Contains swinging motion of a pendulum and a slow rotation motion of the table</li> </ul> <p>Spatial complexity: high</p> <ul style="list-style-type: none"> <li>-Contains complex textures on cactus, shell, flower petals, and golden cloth</li> <li>-Background slightly out of focus</li> </ul>
MBC Test Seq.		<p>Motion level: low</p> <ul style="list-style-type: none"> <li>-Contains a static scene, where a few yellow ribbons on the tree flaps occasionally</li> </ul> <p>Spatial complexity: low</p> <ul style="list-style-type: none"> <li>-Contains mostly flat regions with a few distinctive edges around tree, ribbons, and letters</li> </ul>
Tears of Steel [32]		<p>Motion level: medium</p> <ul style="list-style-type: none"> <li>-Contains simple motions from two people having a conversation</li> </ul> <p>Spatial complexity: medium</p> <ul style="list-style-type: none"> <li>-Contains a few detailed textures on the clothes, hairs, and skin of the people</li> <li>-Background blurred heavily</li> </ul>

## 5 Experimental results

In this section, we present the performance comparison results of the proposed SSSR method and various other state-of-the-art SR methods. For performance evaluation on videos, we used four UHD resolution test sequences with different motions and texture characteristics. The test sequences are 10-s segment videos from European Broadcasting Union (EBU) [31], Blender Foundation [32], and Munhwa Broadcasting Corporation (MBC), and the descriptions of the sequences are listed in Table 4.

The test sequences were first downscaled to FHD resolution and then were upsampled back to UHD resolution using various upscaling methods suitable for high-resolution video applications. The upscaling methods investigated here are bicubic interpolation, proposed method, SSSR implementation by Jun et al. [16], multi-frame-based HD-to-UHD video up-converter solution [19], and convolutional neural network-based SR solution [8]. For evaluating the performance, we did not restrict ourselves to traditional

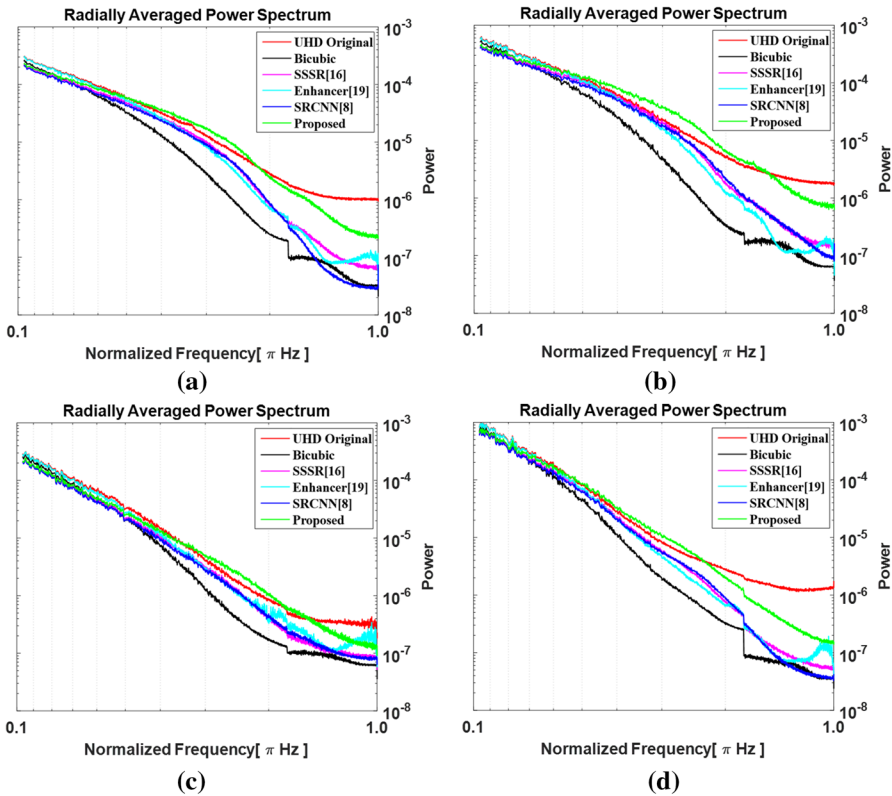


**Fig. 10** Comparison on HR output images (x2 on Pendulus Wide): **a** Original UHD, **b** Bicubic interpolation, **c** SSSR [16], **d** Enhancer [19], **e** SRCNN [8], **f** Proposed

objective metrics, but used measures such as HF reconstruction performance, and subjective quality experiment results to better assess the perceptual quality of the reconstructed videos. Also, to see how proposed method performs on standard test images in objective metric aspect, we compared the PSNR and SSIM values of the proposed method to other state-of-the-art high-complexity SR algorithms [4–9, 11, 12, 20, 21]. The details of the evaluation results and the speed performance of the GPU-based system are provided in following sections. Image examples on various upscaling methods are provided in Fig. 10 and Fig. 13 for visual quality comparison.

### 5.1 High-frequency reconstruction performance

For evaluating the HF signal reconstruction performance of various methods, we used radially averaged power spectrum (RAPS) [33]. The RAPS measures the power magnitude of the frequencies having the same radial distances from the zero frequency and is a convenient way to view and compare the 2D frequency spectrum information in 1D. The average power on each frequency points is calculated as below.



**Fig. 11** Radially averaged power spectrum on various test sequences: **a** Park Dancer, **b** Pendulus Wide, **c** MBC Test Sequence, **d** Tears of Steel

$$P(f_r) = \frac{1}{N(f_r)} \sum_{i \in f_r} \dot{P}(i) \tag{13}$$

The  $f_r$  refers to the radial frequency, which is the sample’s distance from the center zero frequency point in 2D Fourier transform domain. The  $N(f_r)$  refers to the number of discrete frequency samples having radial frequency of  $f_r$ . The  $i$  refers to the set of discrete frequency samples with radial frequency  $f_r$ , and the  $\dot{P}(i)$  is the power of the frequency sample  $i$ . Referring to Fig. 11, the RAPS graph results show that various upscaling methods have similar amount of powers on low frequencies, but tend to diverge on high-frequency area. Throughout various sequences, the bicubic interpolation possesses least amount of HF signals compared to other methods. The SSSR [16], Enhancer [19], and SRCNN [8] presents better HF reconstruction performance when compared to the bicubic interpolation case, but still lacks considerable amount of HF signals when compared to that of the original UHD sequences. The proposed method, on the other hand, not only shows superior HF reconstruction performances when compared to other upscaling methods, but also shows highest fidelity to the HF signals of the UHD original sequences.

### 5.2 Subjective quality evaluation

In order to validate that videos with better HF reconstruction translates to videos with higher perceptual quality, we conducted a subjective experiment based on Double Stimulus Continuous Quality Scale (DSCQS) method recommended by ITU-R BT.500 [34]. The participants are presented with pairs of video sequences, where one is the reference (original) UHD video and the other is the target of evaluation (TOE) video. The participants do not have prior information on which video is the reference. The participants are asked to watch the pair of videos twice and assess the quality of two videos on a continuous scale. The experiment was conducted on 84 in. UHDTV with twenty participants that are nonexperts in image and video processing. Figure 12 shows the result of the subjective experiment in a bar chart form. The red bar indicates the ratio of the case where original UHD video got the higher score, the blue bar indicates the ratio of the case where the TOE video got the higher score, and the gray bar indicates the ratio of the case where the score of the original UHD video and TOE video was the same. Table 5 shows the quality degradation awareness (QDA) ratio results of the various upscaling methods. QDA ratio refers to the ratio of the cases

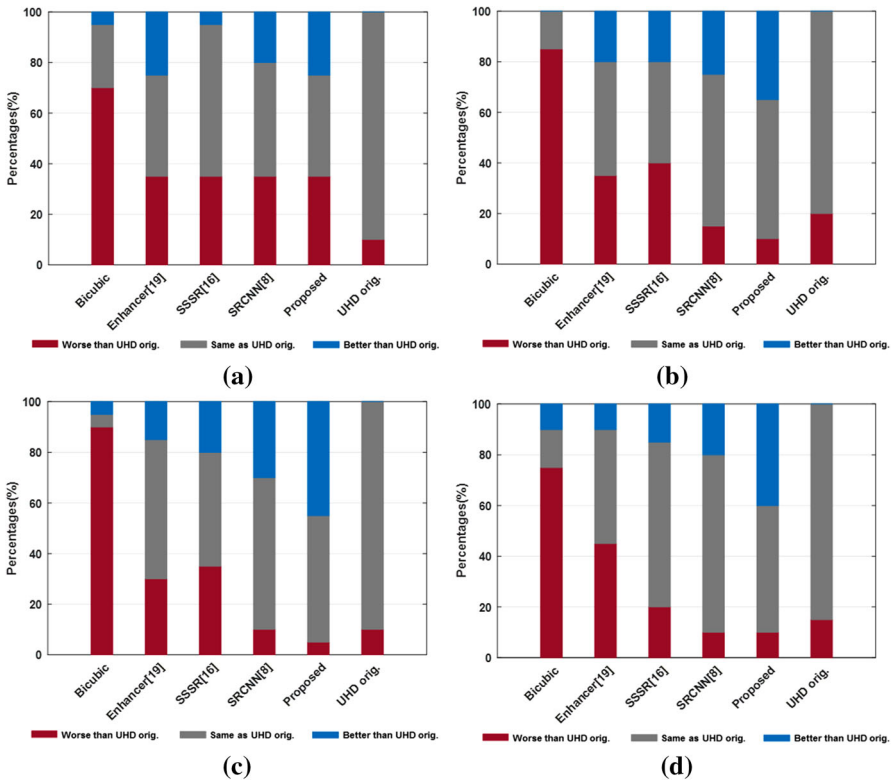


Fig. 12 Subjective experiment results on various test sequences: a Park Dancer, b Pendulus Wide, c MBC Test Sequence, d Tears of Steel



**Table 5** Quality degradation awareness (QDA) ratio results from the subjective experiment - unit in percentages (%)

Videos	Bicubic	Enhancer [19]	SSSR [16]	SRCNN [8]	Proposed	UHD Orig
Park Dancer	70	35	35	35	35	10
Pendulus Wide	85	35	40	15	10	20
MBC Test Seq.	90	30	35	10	5	10
Tears of Steel	75	45	20	10	10	15
Average	80	36.25	32.50	17.50	15	13.75

where TOE videos got lower scores compared to the original UHD video. Higher QDA value indicates that larger number of the participants were able to feel the degraded quality of the TOE video when compared to the original UHD video. As a purpose to investigate the lower bound of the QDA ratio, we measured the QDA value of the original UHD video by including a session where the same original UHD videos are shown twice. The average QDA ratio of this session was 13.75%, so video with QDA value close to 13.75% can be considered to be near the perceptual error bound of the original UHD quality video.

The results in Fig. 12 and Table 5 show similar tendencies to those of the HF reconstruction performance. On various sequences, the bicubic interpolation presents lowest perceptual quality, where 80% of the participants felt the quality were worse compared to the original UHD video. The SSSR [16] and Enhancer [19] present improved subjective quality 32.50 and 36.25% QDA ratio respectively, but are insufficient when compared to the original UHD video quality. The SRCNN [8] and the proposed method, on the other hand, present considerably lower QDA ratio of 17.50 and 15.00% respectively. This means that the SRCNN and the proposed method have higher visual fidelity to the original UHD videos when compared to other methods. One thing to note is that the proposed method has the highest percentage on the case where participants felt the reconstructed video was better than the original UHD video.

### 5.3 Objective metric performance

In this section, we compare the PSNR and SSIM performance of the proposed method to the state-of-the-art algorithms known for their excellent objective performances [4–9, 11, 12, 20, 21]. The measurement was conducted on the standard test images used in the aforementioned literature works. Table 6 shows the objective metric results for various methods. Among the methods, VDSR [12] shows outstanding performance in terms of PSNR where it has 3.66 dB improvement over the bicubic interpolation, but tends to show small improvement in terms of SSIM. This result may be due to the deep layers that are heavily focused on minimizing the average L2-loss rather than preserving the image structures. SRCNN [8] also presents excellent performance, where the PSNR and SSIM improvements over the bicubic interpolation are 3.03 dB and 0.040 respectively. The proposed method presents PSNR and SSIM improvements of 2.02 dB and 0.025 with respect to the bicubic interpolation. Though not as high as the CNN-

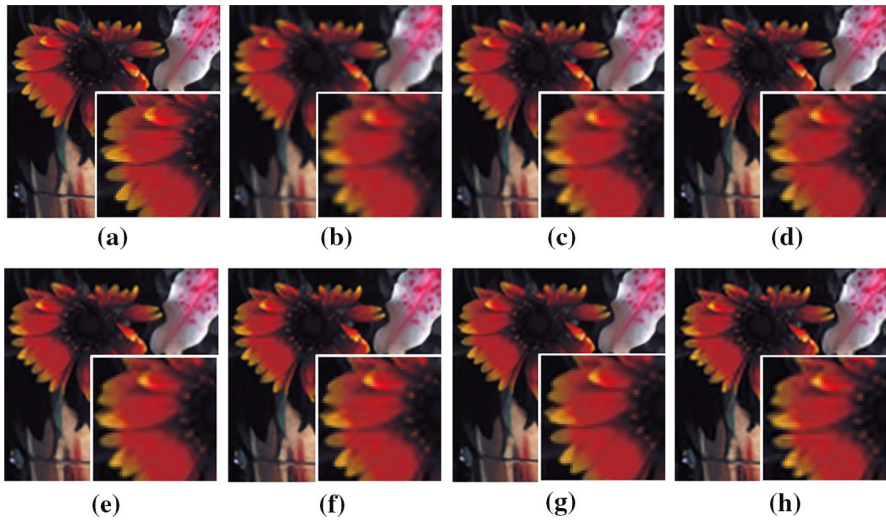


**Table 6** Objective metric results on standard test images: top—PSNR [dB] Bottom—SSIM index

Images	Bicubic	iNEDI [20]	ICBI [21]	NCSR [11]	Zeyde [6]	Peleg [5]
Butterfly	27.44	27.87	27.48	29.08	30.06	30.87
	0.916	0.932	0.925	0.928	0.946	0.956
Comic	26.01	26.53	26.83	26.63	27.44	28.03
	0.850	0.870	0.878	0.861	0.893	0.906
Flowers	30.37	29.84	30.29	30.90	32.02	32.52
	0.899	0.904	0.913	0.891	0.924	0.930
Girl	34.77	34.90	35.34	34.22	35.46	35.59
	0.865	0.884	0.893	0.837	0.883	0.884
Starfish	30.17	29.90	30.52	31.17	31.81	32.81
	0.908	0.893	0.908	0.8874	0.932	0.942
Average	29.75	29.81	30.09	30.40	31.36	31.96
	0.888	0.897	0.903	0.881	0.916	0.924
Images	Yang [4]	APLUS [7]	SRCNN [8]	SI [9]	VDSR [12]	Proposed
Butterfly	31.31	32.00	32.75	31.30	33.84	31.57
	0.960	0.963	0.964	0.956	0.961	0.931
Comic	27.99	28.29	28.52	27.89	29.06	27.43
	0.908	0.916	0.917	0.907	0.919	0.899
Flowers	32.70	33.02	33.32	32.37	33.99	31.90
	0.933	0.936	0.937	0.927	0.920	0.926
Girl	35.63	35.74	35.70	35.49	35.86	35.49
	0.889	0.887	0.886	0.881	0.722	0.883
Starfish	32.73	33.16	33.59	32.68	34.31	32.44
	0.943	0.934	0.937	0.927	0.928	0.926
Average	32.07	32.44	32.78	31.95	33.41	31.77
	0.927	0.927	0.928	0.920	0.890	0.913

based methods [8, 12], the proposed method does present competitive level of objective performances with various state-of-the-art methods. In terms of average PSNR, the proposed method shows similar level of performance with Peleg et al. [5] and SI [9], while outperforming iNEDI [20], ICBI [21], NCSR [11], and Zeyde et al. [6].

Figure 13 shows visual examples of some of the best scoring methods and the proposed method. In the visual quality aspect, SRCNN [8] and the proposed method present best results, where the detailed wrinkles in the red flower petal are properly reconstructed. VDSR [12], APLUS [7], Zeyde [6], and SI [9] methods show some level of perceptual improvements along the outer edge of the red flower petals and the patterns on the white flower petals compared to bicubic interpolation. However, the methods, in general, fail to reconstruct the texture details and the reconstructed images appear blurry. As in this illustration, the proposed method tends to work well on regions with delicate edge and texture details due to its superb HF signal reconstruction capa-



**Fig. 13** Comparison on HR output images (x2 on Flowers): **a** Original, **b** Bicubic interpolation, **c** Zeyde [6], **d** SI [9], **e** APLUS [7], **f** SRCNN [8], **g** VDSR [12], **h** Proposed

**Table 7** Computation time on standard test image up-conversion using CPU resources: time recorded in seconds

Images	iNEDI [20]	ICBI [21]	NCSR [11]	Zeyde [6]	SI [9]	APLUS [7]	Proposed (CPU)
Butterfly	58.19	0.88	159.42	1.07	0.22	0.28	0.14
Comic	107.72	1.06	257.30	1.47	0.28	0.40	0.19
Flowers	142.56	2.33	547.62	3.07	0.62	0.82	0.19
Girl	46.77	0.41	169.25	1.24	0.25	0.33	0.13
Starfish	157.60	1.42	474.79	2.56	0.47	0.69	0.19
Average	102.57	1.22	321.68	1.88	0.37	0.50	0.17

bility. Also unlike SRCNN [8], the proposed method is able to guarantee some level of perceptual sharpness enhancement without requiring an appropriate external database. An important point to note here is that, though the proposed method is focused on providing fast processing time suitable for high-resolution video applications, it is still able to present a level of performance that is comparable to the quality-oriented, computationally complex methods. In next section, we provide the computation time comparison of the various methods and show how the proposed method perform in terms of quality versus computational complexity aspect.

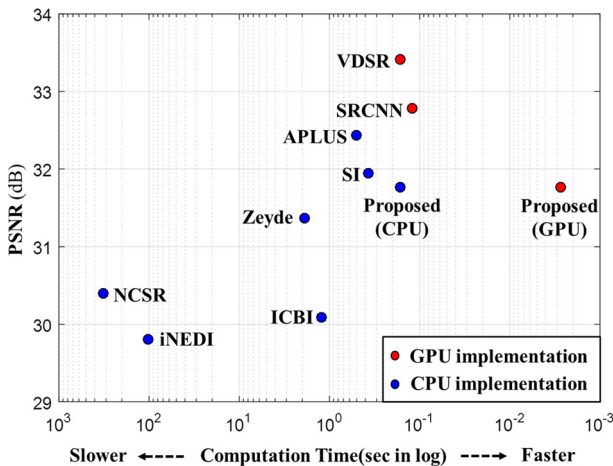
#### 5.4 Speed performance

The speed performances of the proposed method, along with the other state-of-the-art methods [6–9, 11, 12, 16, 20, 21], were investigated on a platform consisting of Intel®

**Table 8** Computation time on standard test image up-conversion using single GPU card: time recorded in seconds

Images	SRCNN [8]	VDSR [12]	Proposed (GPU)
Butterfly	0.16	0.19	0.0024
Comic	0.08	0.11	0.0025
Flowers	0.15	0.21	0.0032
Girl	0.08	0.12	0.0025
Starfish	0.13	0.19	0.0030
Average	0.12	0.16	0.0027

core (TM) i7-5930X CPU @ 3.50 GHz 32 GB RAM with four NVIDIA GTX 980 Ti GPU cards. The CNN-based methods [8, 12] are implemented to utilize GPU resources using NVIDIA cuDNN library, and the rest of the methods utilize CPU resources. For fair comparison, we implemented two versions of the proposed method, one which only utilizes the CPU resources and the other that utilizes GPU resources. Then, we compared each version of the proposed method with other methods that use same kind of computation resources. Tables 7 and 8 show the computation time results on standard test images using CPU and GPU resources, respectively. The results on Table 8 are the results when using only single GPU card. Figure 14 shows the performance comparison on the standard test images in terms of quality versus computation time. The results show that proposed method presents competitive quality performance, while having significantly faster processing time compared to other methods. When using CPU resources, the proposed method has similar level of PSNR with Zeyde et al. [6] and SI [9], while having a speed performance that is approximately 11 times and 2 times faster, respectively. When using GPU resources, the proposed method has similar level of perceptual quality compared to SRCNN [8] and VDSR [12], while having a



**Fig. 14** Performance comparison in terms of quality versus computation time for the proposed method and other state-of-the-art methods with standard test images. The PSNR values of the methods are provided in Table 6, and the computation time results are provided in Tables 7 and 8

**Table 9** Computation time on FHD-to-UHD video up-conversion using multiple (four) GPU cards

Videos	No. of frames	SRCNN [8]		VDSR [12]		Proposed (GPU)	
		Computation time (s)	Conversion speed (fps)	Computation time (s)	Conversion speed (fps)	Computation time (s)	Conversion speed (fps)
Park Dancer	500	92.51	5.41	348.02	1.44	8.27	60.47
Pendulus Wide	500	93.01	5.38	349.52	1.43	8.28	60.36
MBC Test Seq.	600	111.01	5.41	421.22	1.42	9.79	61.31
Tears of Steel	240	44.16	5.43	168.25	1.43	4.01	59.91
Average		85.17	5.41	321.75	1.43	7.59	60.51

speed performance that is approximately 40 times and 60 times faster, respectively. The speedup ratio of the proposed method from converting the CPU implementation to GPU implementation is approximately 60 times.

In order to evaluate the speed performance of the SR methods on multiple GPU cards, we set the proposed method, SRCNN [8] and VDSR [12] to perform FHD-to-UHD video up-conversion using all four GPU cards on the platform. Table 9 shows the total computation time and the average fps for converting each sequences. As shown in the table, the proposed method provides a speed performance that is approximately 10 times and 40 times faster than SRCNN [8] and VDSR [12], respectively. It is worth noting that the proposed method implemented on a single PC with four GPU cards is able to provide average conversion speed of over 60 fps. This means that the proposed method can provide real-time conversion speed for 60 Hz FHD videos on a single platform.

Considering that the proposed method is a single-frame-based method that has no dependencies between the data processed by each GPU cards, the conversion speed can be further accelerated by increasing the number of GPU cards in the system.

## 6 Conclusion

In this paper, we propose a super-resolution method with region-adaptive HF enhancement algorithm. The proposed method first reconstructs HF signals from a self-similar region within a frame, and then adaptively enhances the HF signals with different enhancement factors based on the difference curvature region classification. The proposed method is able to improve the perceptual sharpness through HF signal enhancement, while also minimize the possible visual quality degradations by adjusting the enhancement factors on the regions where the artifacts are likely to be noticeable. The experimental results show that the proposed method not only has a superior HF reconstruction performance compared to other competitive upscaling solutions, but also produces output videos that are perceptually as sharp as the original high-resolution videos. The proposed method was implemented on a system with multiple GPUs using OpenCL framework. The experimental results show that the system is able to provide real-time conversion speed and can be further accelerated by increasing the number of GPU card in the system. Due to its fast and high-quality up-conversion capability, the proposed system can be practically applied on various consumer products such as UHDTV, surveillance system, and mobile devices.

**Acknowledgements** This work was supported by ICT R&D program of MSIP/IITP. [B0101-16-1280, Development of Cloud Computing Based Realistic Media Production Technology].

## References

1. Hardie RC, Barnard KJ, Armstrong EA (1997) Joint MAP registration and high-resolution image estimation using a sequence of undersampled images. *IEEE Trans Image Process* 6:1621–1633
2. Farsiu S, Robinson MD, Elad M, Milanfar P (2004) Fast and robust multiframe super-resolution. *IEEE Trans Image Process* 13:1327–1344

3. Ng MK, Shen H, Lam EY, Zhang L (2007) A total variation regularization based super-resolution reconstruction algorithm for digital video. *EURASIP J Adv Signal Process* 2007:74585-1-74585-16
4. Yang J, Wang Z, Lin Z, Cohen S, Huang T (2012) Coupled dictionary training for image super-resolution. *IEEE Trans Image Process* 21(8):3467-3478
5. Peleg T, Elad M (2014) A statistical prediction model based on sparse representations for single image super-resolution. *IEEE Trans Image Process* 23(6):2569-2582
6. Zeyde R, Elad M, Protter M (2010) On single image scale-up using sparse-representations. In: *Curves and surfaces*, pp 711-730
7. Timofte R, De Smet V, Van Gool L (2014) A+: adjusted anchored neighborhood regression for fast super-resolution. In: *Proceedings of Asian Conference on Computer Vision*, Singapore
8. Dong C, Loy CC, He K, Tang X (2015) Image super-resolution using deep convolutional networks. *IEEE Trans Pattern Anal Mach Intell* 38:295-307
9. Choi J-S, Kim M (2016) Super-interpolation with edge-orientation-based mapping kernels for low complex upscaling. *IEEE Trans Image Process* 25:469-483
10. Dong W, Zhang L, Shi G, Wu X (2011) Image deblurring and superresolution by adaptive sparse domain selection and adaptive regularization. *IEEE Trans Image Process* 20(7):1838-1857
11. Dong W, Zhang L, Shi G, Li X (2013) Nonlocally centralized sparse representation for image restoration. *IEEE Trans Image Process* 22(4):1620-1630
12. Kim J, Kwon Lee J, Mu Lee K (2016) Accurate image super-resolution using very deep convolutional networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*
13. Glasner D, Bagon S, Irani M (2009) Super-resolution from a single image. In: *12th International Conference on Computer Vision*, pp 349-356
14. Freedman G, Fattal R (2011) Image and video upscaling from local self-examples. *ACM Trans Graph* 30, Article No. 12, April 2011
15. Yang J, Lin Z, Cohen S (2013) Fast image super-resolution based on in-place example regression. In: *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, 23-28 June 2013, pp 1059-1066
16. Jun JH, Choi JH, Lee DY, Jeong S, Cho SH, Kim HY, Kim JO (2015) Accelerating self-similarity-based image super-resolution using OpenCL. *IEIE Trans Smart Process Comput* 4(1):10-15
17. Chen S, Gong H, Li C (2011) Super resolution from a single image based on self-similarity. In: *IEEE International Conference on Computational and Information Sciences*, Oct 2011, pp 91-94
18. Khronos OpenCL Working Group (2012) The OpenCL Specification Version 1.2. Khronos Group. <http://www.khronos.org/opencl>
19. Version 2.1.0, July 2016. <http://www.infognition.com/videoenhancer/>
20. Asuni N, Giachetti A (2008) Accuracy improvements and artifacts removal in edge based image interpolation. In: *Proceedings Third International Conference Computer Vision Theory and Applications (VISAPP'08)*
21. Giachetti A, Asuni N (2011) Real-time artifact-free image upscaling. *IEEE Trans Image Process* 20(10):2760-2768
22. Yang S, Kim Y, Jeong J (2008) Fine edge-preserving technique for display devices. *IEEE Trans Consum Electron* 54(4):1761-1769
23. Kang W et al (2013) Real-time super-resolution for digital zooming using finite kernel-based edge orientation estimation and truncated image restoration. In: *Proceedings of 20th IEEE International Conference on Image Processing*, Melbourne, VIC, Sept 2013, pp 1311-1315
24. Park SJ, Lee OY, Kim JO (2013) Self-similarity based image super-resolution on frequency domain. In: *Proceedings of APSIPA ASC 2014*, Nov 2013, pp 1-4
25. Wang Z, Bovik AC, Evans BI (2000) Blind measurement of blocking artifacts in images. In: *IEEE International Conference on Image Processing*, Sep 2000, pp 981-984
26. Bae S-H, Kim M (2014) A novel generalized DCT-based JND profile based on an elaborate CM-JND model for variable block-sized transforms in monochrome images. *IEEE Trans Image Process* 23(8):3227-3240
27. Alam MM (2014) Local masking in natural images: a database and analysis. *J Vis* 14(8):22-22
28. Ponomarenko N et al (2013) Color image database TID2013: peculiarities and preliminary results. In: *Proceedings of 4th European Workshop on Visual Information Processing*, June 2013, pp 106-111

29. Song L, Tang X, Zhang W, Yang X, Xia P (2013) The SJTU 4K video sequence dataset. In: The Fifth International Workshop on Quality of Multimedia Experience (QoMEX2013), Klagenfurt, 3rd–5th July 2013
30. Chen Q, Montesinos P, Sun Q, Heng P, Xia D (2010) Adaptive total variation denoising based on difference curvature. *Image Comput* 28(3):298–306
31. European Broadcast Union (2013) EBU UHD-1 Test Set (online). <http://tech.ebu.ch/testsequences/uhd-1>
32. Blender Foundation (2012) Tears of Steel, Mango Open Movie Project. <http://tearsofsteel.org>
33. Ulichney RA (1988) Dithering with blue noise. *Proc IEEE* 76(1):56–79
34. ITU-R BT.500-13 (2012) ITU, Methodology for the subjective assessment of the quality of television pictures