DOI: 10.2352/ISSN.2470-1173.2016.18.DPMI-028

Light-Weight Single Image Super-Resolution via Pattern-wise Regression Function

Kohei Kurihara[a], Yoshitaka Toyoda[a], Shotaro Moriya[b], Daisuke Suzuki[a], Takeo Fujita[a], Narihiro Matoba[a], Jay E. Thorton[c], Fatih Porikli[d]

[a] Mitsubishi Electric Corporation, Advanced Technology R&D Center, 1 Zusho Baba Nagaokakyo-shi, Kyoto, Japan

[b] Mitsubishi Electric Corporation, Kamakura Factory, 325 Kamimatiya Kamakura-shi, Kanagawa, Japan

[c] Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA.

[d] Australian National University (ANU), Canberra ACT 0200 Australia National ICT Australia (NICTA) Australia

Abstract

We propose a novel upsampling approach that is suitable for hardware implementation. Compared with past super-resolution (SR) upsampling methods (e.g. example based upsampling), structure of our upsampling approach is very simple. Strategy of our approach is mainly 2 terms; off-line training term and real-time upscaling term. (i)During training term, grouping low-resolution (LR) - high-resolution (HR) patch pairs and determined a linear regression function of each groups. And (ii)during upscaling term, assigning pattern number to each of input LR patches according to the signature using a local binary pattern (LBP), and transforming input LR patches to HR patches by applying the trained regression function based on the LBP in a patch-by-patch fashion. Our evaluation result shows that our method is comparable to other state-of-the-art methods. Furthermore, our approach is compactly implemented on LSI (e.g. FPGAs) or be shorten the processing time on software because of simplicity of the structure.

Introduction

Currently, security systems become an increased center of focus in our daily life. There is a demand for recognizing some distant objects such as faces or license plates on a big display. In order to do that, we upscale low-resolution (LR) images to high resolution (HR) images. And the transformation poses a problem because the new image should have more information of pixels than the original image.

A simple method of increasing the image resolution is interpolation technique, such as bilinear and bicubic interpolation. Although these methods are quite simple to implement, these approaches cannot estimate and reconstruct high-frequency information in generated HR image, because these methods assumed image smoothness.

Super-Resolution is another technique of upsampling method which generates an HR image from multiple LR images [1] or single LR image [2][3][4]. This upsampling is always ill-posed problem because amount of pixels in target image is typically higher than those in observed input image(s). Conventional SR methods use multiple images with small motion as inputs. Although those methods need a mechanics of moving camera system(e.g. aerial images), this approach can reconstruct latent high-frequency elements.

Another type of SR technique uses database constructed by training. This approach typically learns a relationship between the HR patches and the downsampled LR patches in advance. The example based upsampling method [2][3] is one of those approach and it can estimate high-frequency information.

It is known that the example based method has two problems. One is that this method should have a large amount of database in order to deal with large variation of real world scenes. This large variation of database requires huge computational resources and it is not suitable for embedded systems. The other problem is that resulting images sometimes include noise, halos, ringing, and aliasing artifacts, because of mismatch between the input image and the image database.

Our goal is to generate high-resolution, artifact-less output images using reasonable computational resources. In Section 2 and 3, we describe our proposed method. In Section 4 we demonstrate the experimental results of our method compared with several SR methods. And finally in Section 5 we conclude the paper.

Super Resolution via Pattern-wise Regression Function

In this section, we describe our super-resolution algorithm. Our method generates a HR image from a LR image using linear regression functions. The method uses 2 main stages; off-line training term and real-time upscaling term.

Data Training

During training term (Fig.1.), our approach groups LR-HR patch pairs of similar pattern according to the signature using a local binary pattern (LBP). In each of the group, we determine a linear regression function that will estimate an output HR patch corresponding to the LR patch.

First, LR image is obtained by downsamping the HR image. This LR image is used for discriminating LBP, which can determine the signature of each local area of input image (explained in latter section). Next, in order to group LR-HR patch, we upscale the downsampled LR image and obtain the same size, but low resolution compared with HR image. Then upscaled LR patch and corresponding HR patch are extracted (in this case, size of a patch is 5×5 pixels). To eliminate the influence of a luminance offset of image patch, texture component and non-texture component (average luminance of a patch) are separated from the patch and texture components of LR and HR patches are used for making patch-pair. LR-HR patch pairs of all pixels are categorized into 128 groups on the basis of pattern number *Pnum*. To approximate a relationship between LR patch and HR patch, we determine a linear regression function that will estimate an output HR patch. Coefficient data M (25×25) is calculated according to below regression function equation. Then, X_t / Y_t indicates a matrix of accumulated LR / HR texture patches in 1 LBP group. *eye* is an unit matrix and λ is a static parameter. LR patch.

DOI: 10.2352/ISSN.2470-1173.2016.18.DPMI-028

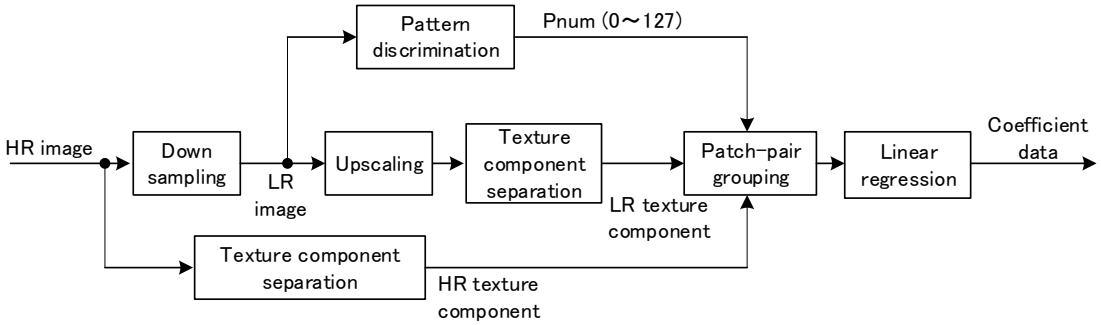

Fig.1. Flowchart of our proposed algorithm (off-line training term)

$$M = (X_t^{'} \cdot X_t + \lambda \cdot eye(5\times5, 5\times5))^{-1} \cdot X_t^{'} \cdot Y_t \quad (1)$$

For data reduction and small calculation in upscaling term, only elements in coefficient data corresponded to center area (3×3 pixels) of image patch is extracted. Therefore size of all accumulated coefficient data is 128×25×9. They are used for estimating accurate HR patch from LR patch in upscaling term.

Upscaling

During real-time upscaling term (Fig.2.), the HR image is generated from the input LR image by applying the trained regression function based on the LBP in a patch-by-patch fashion. LR image patch (5×5 pixels) is extracted from upscaled LR image at intervals of 2 pixels. Then LR patch is separated into texture and non-texture component (average luminance of a patch). On the other hand, LR image patch (3×3 pixels) which is used for pattern discrimination is extracted from input LR image (hereafter called pattern patch). As described in Fig.3, center pixel of 3×3 pixels upscaled patch and that of pattern patch is same position.

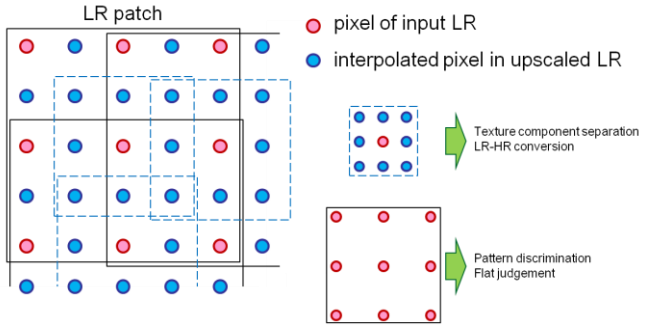

Fig.3. Extracting patches

LBPs are discriminated from pattern patch by applying same fashion used in training term. Also flat judgment (described in Section 3) is performed to pattern patch. Then coefficient data (25×9) corresponded to the LBP is selected from 128 groups and it is used for LR-HR patch transformation. Texture component of LR patch is performed matrix multiplication with coefficient data for transforming texture component of LR patch to that of HR patch. For detail enhancement of output image, texture component of HR patch can be adjusted by multiplying scalar gain (called Enhance Gain). HR patch is recomposed by adding texture component of HR patch and non-texture component of LR patch. Finally all HR patches are merged, then overlapped pixels are averaged, and we obtain HR image. We perform post smoothing filter (described in Section 3) to HR image with upscaled LR image as a guide, we obtain final output HR image.

Pattern Discrimination using LBP

Pattern discrimination is performed on LR patch (3×3 pixels) which centers each pixel of LR image and these patches are extracted through all pixels of downsampled LR image.

Fig.4. describes a way to calculate pattern number *LBPnum*. In order to obtain LBP, each LR patch is performed binarization. And a pixel value of binarized patch *b* assigns 0 when corresponded pixel in LR patch is smaller than *AVE* (weighted average of pixels in a patch), otherwise assigns 1. The reason of using weighted average *AVE* instead of using a center pixel value is to increase noise robustness of pattern discrimination.

$$b(s,t) = \begin{cases} 0 & when \quad LR(s,t) < AVE \\ 1 & when \quad LR(s,t) \geq AVE \end{cases} \quad (2)$$

The *LBPnum* is derived from combination of LBP (Refer Fig.4.).

$$\begin{aligned} LBPnum = {} & LBP(-1,-1)\cdot 2^0 + LBP(-1,0)\cdot 2^1 + LBP(-1,1)\cdot 2^2 \\ & + LBP(0,1)\cdot 2^3 + LBP(1,1)\cdot 2^4 + LBP(1,0)\cdot 2^5 \\ & + LBP(1,-1)\cdot 2^6 + LBP(0,-1)\cdot 2^7 \end{aligned} \quad (3)$$

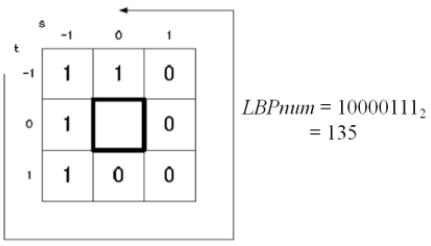

Fig.4. Calculation of *LBPnum*

Additionally, in order to decrease a number of patterns (256→128), a pair of LBPs whose patterns are same but inversed are considered as same LBP.

$$Pnum = \min(LBPnum, 255 - LBPnum) \quad (4)$$

$$LBPsym = \begin{cases} 0 & when \quad Pnum = LBPnum \\ 1 & when \quad Pnum = 255 - LBPnum \end{cases} \quad (5)$$

DOI: 10.2352/ISSN.2470-1173.2016.18.DPMI-028

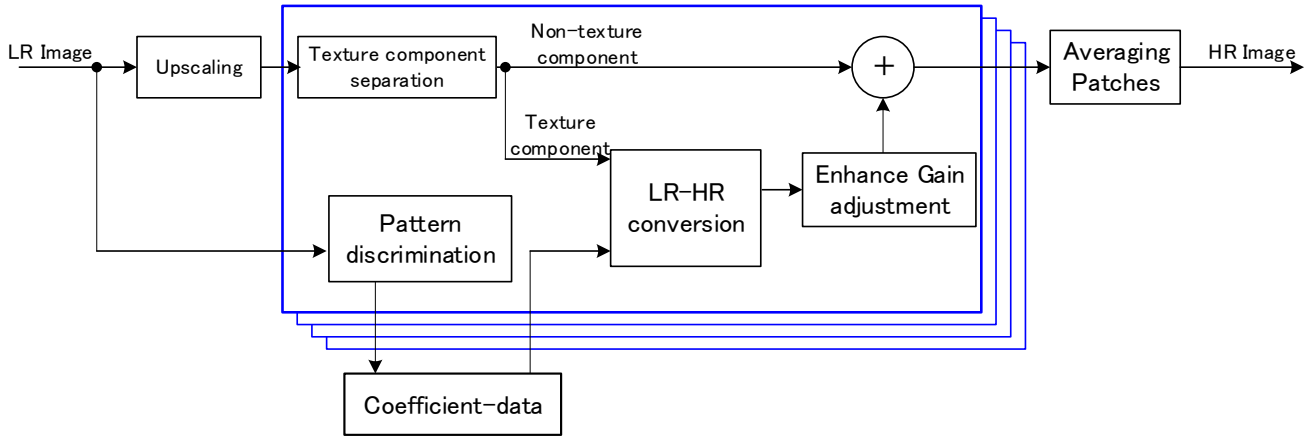

Fig.2. Flowchart of our proposed algorithm (real-time upscaling term)

Grouping LR-HR patch pairs is based on *Pnum*, which indicates the signature of LR and HR patch.

Note that this upscaling method is quite easy to implement because identifying LBP of each LR patch is used a simple binarization. And LR-HR transformation is also a simple linear combination (multiply-add operation) of patch and regression function.

Additional improvement processing

Additionally, in order to improve the upscaling performance, our method applies flat judgment and original post-smoothing filter.

Flat judgment

Flat judgment checks whether patch of interest is smooth or not. And if the patch is smooth, SR method does not apply SR upscaling to avoid unwanted emphasis of noises on a flat area. When *diff,* which is an average of the difference between LR patch and its luminance offset *AVE*, is smaller than preset threshold, LR-HR patch transformation is not performed.

$$diff = \frac{1}{8} \times \sum_{j=-1}^{+1} \sum_{i=-1}^{+1} |LR(i,j) - AVE| \tag{6}$$

$$\begin{aligned} AVE = & \frac{LR(-1,-1)}{16} + \frac{LR(0,-1)}{8} + \frac{LR(1,-1)}{16} \\ & + \frac{LR(-1,0)}{8} + \frac{LR(0,0)}{4} + \frac{LR(1,0)}{8} \\ & + \frac{LR(-1,1)}{16} + \frac{LR(0,1)}{8} + \frac{LR(1,1)}{16} \end{aligned} \tag{7}$$

Post smoothing filter

Post-smoothing filter applies edge directed sorting filter for resulting SR image by using simply upsampled LR image as a guide. Although structure of this filter is very simple, the method effectively inhibits noises while preserving edges.

This process is performed for all pixels *(x, y)* in SR image. First, 3×3 pixels patch is extracted from upscaled LR image. Then differences between center pixel of the patch and its peripheral 8 pixels are calculated. The position of pixels (x^{1st}, y^{1st}), (x^{2nd}, y^{2nd}), corresponded to the most and the second smallest difference values, might indicate edge direction (as described in Fig.5). It is important that this estimation of edge direction is not affected by generated noise and artifact included in SR image, because this

$$\begin{aligned} OUT(x,y) = & (1-\alpha) \times SR(x,y) + \\ & \alpha \times \{SR(x^{1st}, y^{1st}) + SR(x^{2nd}, y^{2nd})\} \end{aligned} \tag{8}$$

procedure refers upscaled LR image, not SR image. A pixel *OUT(x, y)*, pixel of final filtered output image, is a weighted sum of the pixel *SR(x, y)* in SR image and average of the pixels SR(x^{1st}, y^{1st}), and *SR* (x^{2nd}, y^{2nd}).

α is a static parameter which controls a blend ratio of the filtered and the unfiltered images. Compared with ordinal low-pass filter (e.g. box filter), this filter can preserve sharpness and smooth noise near edges simultaneously.

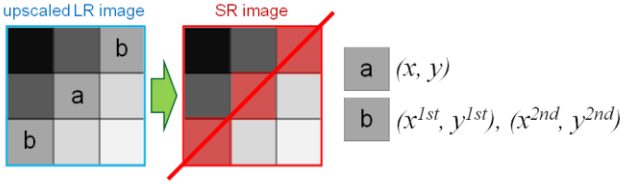

Fig.5. processing image of post smoothing filter

Experiment

In this section we analyze the performance of our proposed method by applying quantitative and qualitative comparison with proposed method and other state-of-the-art methods.

Experimental Settings

Our proposed algorithm was implemented in MATLAB R2012b and tested. The training HR image data was carefully selected from our original dataset (e.g. humans, characters, texture, etc.) taken by digital still camera (Canon 5D mark II). Note that in our method, size of coefficient data is independent with the amount of training data. The reason is because, regardless of the number of training patches, they are always categorized into 128 groups. And therefore that of coefficient data generated from linear regression is static. We compare our proposed method and other methods in terms of PSNR (dB) and SSIM with magnification factor of 4.

We chose 4 other upsampling methods (bicubic, NEDI[4], ScSR[5], and ANR[6]) for comparison, and used 2 datasets (standard test image dataset, and our original dataset described in Fig.6.) for test images. Note that the training image data does not include test images.

DOI: 10.2352/ISSN.2470-1173.2016.18.DPMI-028

Table1. PSNR (dB) and SSIM for upscaling factor ×4

Image Name	Bicubic		NEDI		ScSR		ANR		proposed	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
baby	30.37	0.917	29.30	0.892	31.04	0.934	**31.46**	**0.937**	31.26	0.930
baboon	20.80	0.697	20.61	0.663	20.98	**0.742**	**21.00**	0.741	20.95	0.736
barbara	23.37	0.795	23.11	0.782	23.51	0.811	**23.54**	**0.812**	**23.54**	0.810
lenna	28.27	0.902	28.00	0.890	28.79	0.915	**29.13**	**0.919**	29.05	0.914
butterfly	24.73	0.928	24.06	0.912	25.50	0.940	25.92	**0.946**	**25.95**	**0.946**
pepper	28.61	0.928	27.94	0.917	29.03	0.930	**29.44**	**0.935**	29.38	0.932
port	23.37	0.839	22.60	0.805	23.98	0.868	**24.16**	**0.873**	24.14	0.870
calender	22.89	0.750	22.49	0.719	23.27	0.791	23.33	0.796	**23.37**	**0.801**
girl	33.34	0.949	33.40	0.948	33.98	0.954	34.68	**0.958**	**34.71**	0.953
kimono	27.79	0.884	27.03	0.863	28.40	0.902	**28.73**	**0.905**	28.60	0.895
temple	19.21	0.685	18.78	0.629	19.48	**0.728**	**19.51**	0.727	19.50	0.727
bill	28.76	0.878	28.53	0.863	29.61	0.896	30.19	**0.901**	**30.28**	0.895
average	25.96	0.846	25.49	0.824	26.46	0.868	**26.76**	**0.871**	26.73	0.867

In our method, although enhance gain is useful for increasing image visibility, detail enhancement may cause bad result for full reference objective assessment like PSNR. Therefore in this experiment we set to no enhancement.

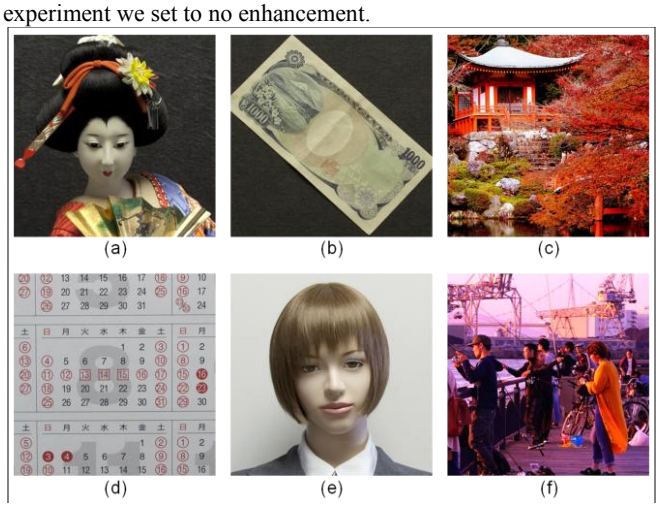

Fig.6. original dataset for test images of SR
(a)kimono, (b)bill, (c)temple, (d)calendar, (e)girl, (f)port

Experimental Result

In this section we will show quantitative and qualitative results. First, we demonstrate the quantitative results of 4 × magnification. Table 1 summarizes the results, showing PSNR and SSIM values for a number of test images.

Compared with ScSR (one of the representative SR technique based on training approach), our method has good performance in PSNR and almost same performance in SSIM. (Table1, 2) Although PSNR and SSIM performance of ANR method is slightly higher than that of our proposed method, our approach is able to compete with the current state-of-the-art image upsampling methods. Also visual examples are shown in Fig.7, 8, and 9. Focusing on slanted edges (e.g. stripes in butterfly, wheels of a bicycle), those of our method are smooth and less jaggy. And amount of noise and artifact in our proposed method is as less as those of ANR result.

Hardware Implementation

Next, in order to examine an implementation suitability of our proposed method, we embedded our algorithm into FPGA. Brief specification of that FPGA is indicated in Table 2.

Table2. Brief specification of the FPGA

upscaling factor	× 1.25 ～× 4
input image size	128 × 64 pixels ～ 1920 × 1080 pixels
frame rate	**30 fps**
frame delay	＜ 1frame

Our FPGA can generate upscaled (magnification factor is ×1.25 to ×4) Full HD (1920 × 1080 pixels) image sequence at 30 fps. And circuit size of this method is small enough to be embedded in system LSI as one function.

In actual case, for instance our SR method is applied to digital cameras, we adjust parameters of SR processing to sharper setting. The purpose of this adjustment is to increase a visibility of target object (e.g. license plates, faces). A visual example of the use case image (LabelMe [7]) is shown in Fig.10. Qualitatively, compared with a result of ordinal bicubic interpolation, at least in number recognition aspect, result of our method looks superior. Moreover, our result with real case condition looks slightly better than the result with default condition (although real case condition might perform bad result for full reference objective assessment). For some situation and purpose, this proposed method can be optimized.

DOI: 10.2352/ISSN.2470-1173.2016.18.DPMI-028

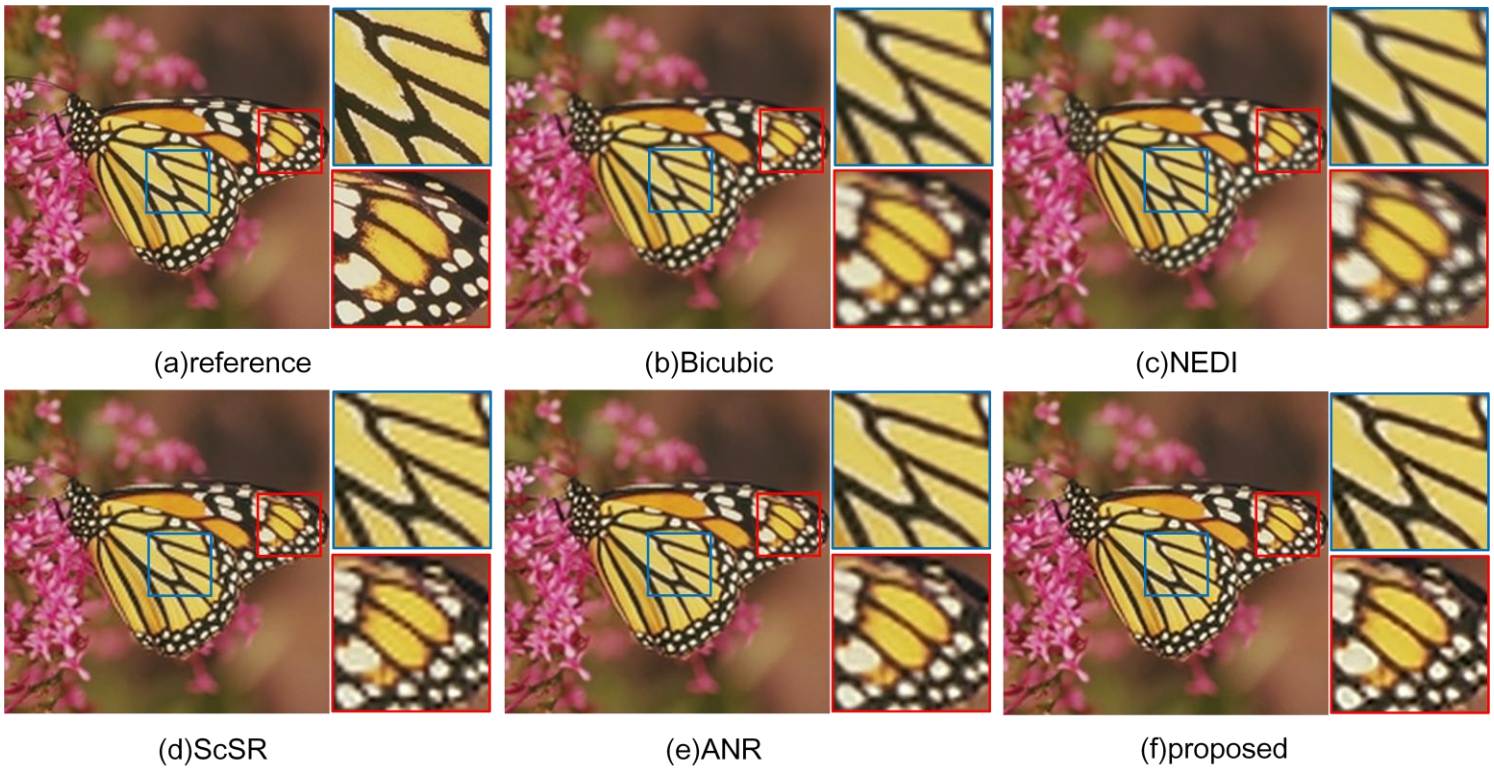

Fig.7. Visual qualitative assessment for “butterfly” image with magnification×4

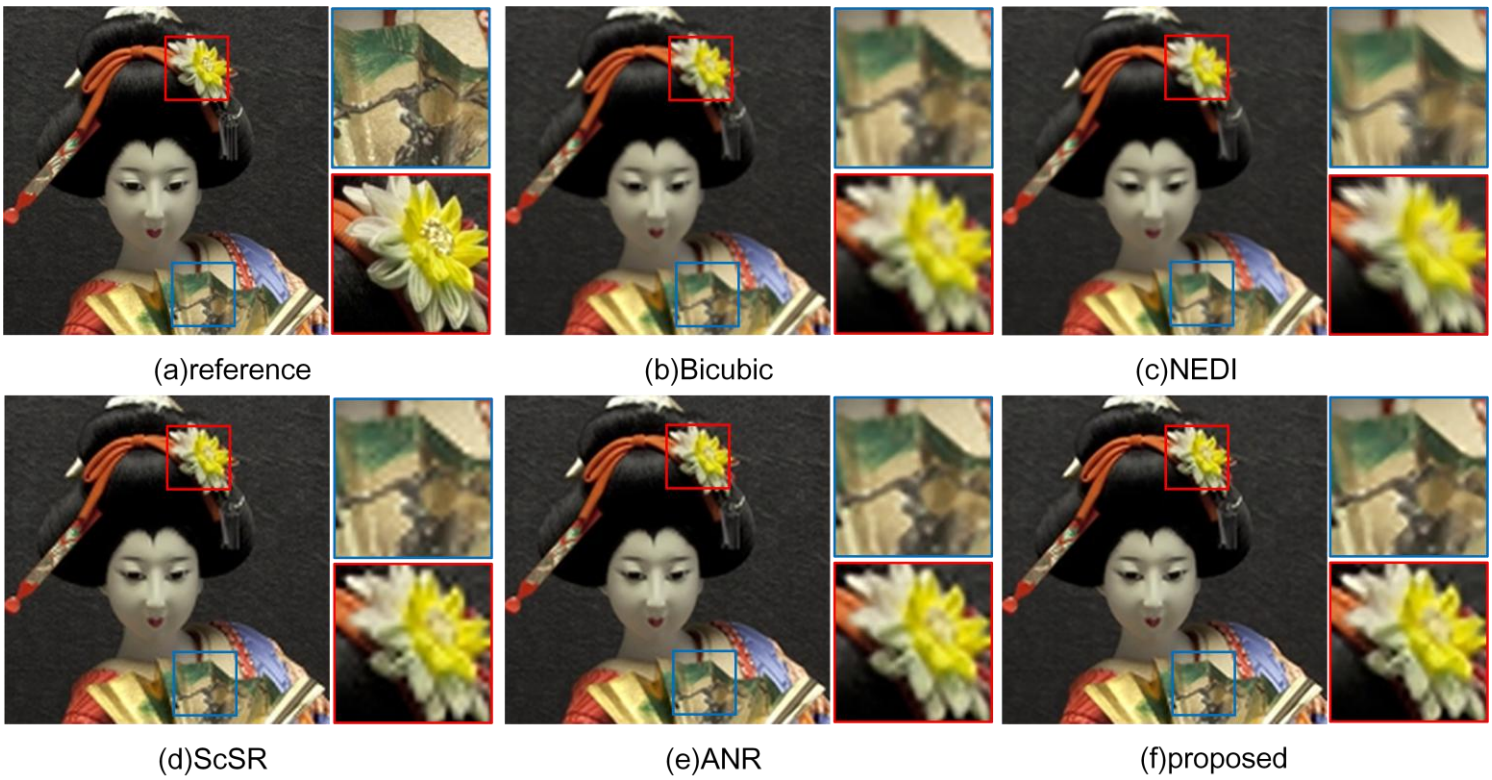

Fig.8. Visual qualitative assessment for “kimono” image with magnification×4

DOI: 10.2352/ISSN.2470-1173.2016.18.DPMI-028

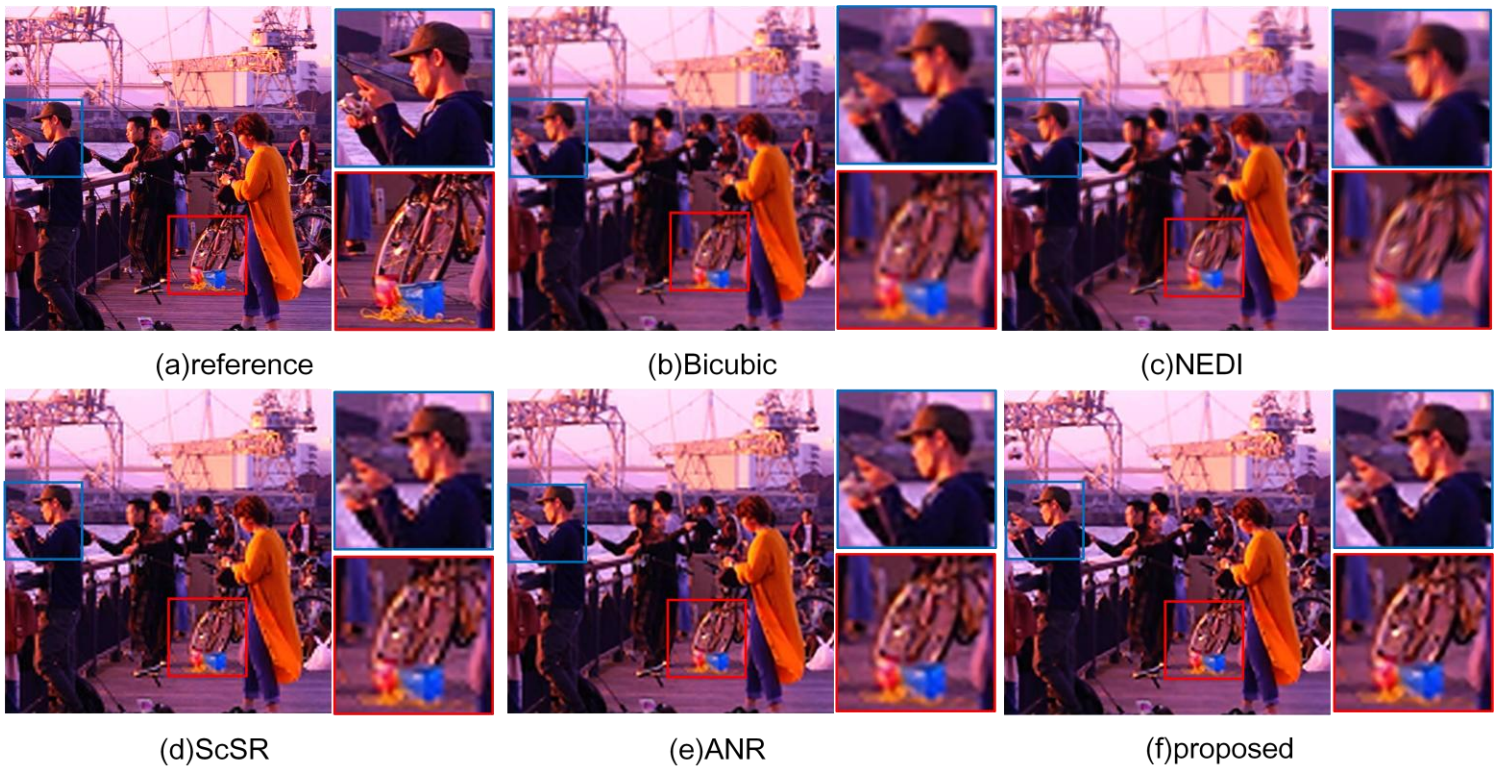

Fig.9. Visual qualitative assessment for "port" image with magnification×4

Fig.10. Visual qualitative comparison of objective visibility with magnification×4

DOI: 10.2352/ISSN.2470-1173.2016.18.DPMI-028

Conclusion

Performance of our proposed approach is comparable to other related state-of-the-art methods. Furthermore, our approach is compactly implemented on LSI (e.g. FPGAs) or be shorten the processing time on software because of simplicity of the structure.

LSI implementation of our method enables us to obtain high-quality HR image in real-time action. Therefore we achieved to generate high-resolution, artifact-less output images using reasonnable computational resources.

References

[1] S. Farsiu, D. Robinson, M. Elad, P. Milanfar. 2003a. Fast and robust Super-Resolution. Proc 2003 IEEE Int Conf on Image Process 2003, pp.291–294.

[2] W. T. Freeman, T. R. Jones, and E. C. Pasztor. Example based super-resolution. Comp. Graph. Appl., (2), 2002.

[3] Glasner, D., Bagon, S., and Irani, M., "Super-resolution from a single image," in [IEEE International Conference on Computer Vision], 349{356, IEEE (2009).

[4] Li, X., Orchard, M.T.: New edge-directed interpolation. IEEE Trans. Image Processing 10, 1521–1527 (2001)

[5] Yang, J., Wright, J., Huang, T. S., Ma Y.: Image super-resolution as sparse representation of raw image patches. CVPR (2008) 1-8

[6] Timofte, R., De Smet, V., Van Gool, L.: Anchored Neighborhood Regression for Fast Example-Based Super Resolution. ICCV (2013) 1920-1927

[7] B. C. Russell, A. Torralba, K. P. Murphy, W. T. Freeman, LabelMe: a database and web-based tool for image annotation. International Journal of Computer Vision, pages 157-173, Volume 77, Numbers 1-3, May, 2008.

Author Biography

Kohei Kurihara received B.E. and M.E. degrees from Osaka University, Osaka, Japan, in 2011 and 2013. He joined Mitsubishi Electric Corporation in 2013, and has been engaged in the research and development of image processing for cameras.