International Journal of Innovative
Computing, Information and Control
Volume 9, Number 2, February 2013

ICIC International ©2013 ISSN 1349-4198
pp. 903–914

FAST IMAGE-ENLARGEMENT ALGORITHM FOR THE AUGMENTATION OF THE HIGH-FREQUENCY COMPONENT BY EMPLOYING A HIERARCHICAL PREDEFINED CODEBOOK

Hakaru Tamukoh[1], Hideaki Kawano[2], Noriaki Suetake[3]
Byungki Cha[4] and Takashi Aso[4]

[1]Institute of Technology
Tokyo University of Agriculture and Technology
2-24-16, Naka-chou, Koganei-shi, Tokyo 184-8588, Japan
tamukoh@cc.tuat.ac.jp

[2]Faculty of Engineering
Kyushu Institute of Technology
1-1, Sensui-cho, Tobata-ku, Kitakyushu 804-8550, Japan

[3]Graduate School of Science and Engineering
Yamaguchi University
1677-1, Yoshida, Yamaguchi-shi, Yamaguchi 753-8512, Japan

[4]Faculty of Management and Information Sciences
Kyushu Institute of Information Sciences
6-3-1, Saifu, Dazaifu-shi, Fukuoka 818-0117, Japan

Received December 2011; revised May 2012

Abstract. *High-quality image enlargement is achieved by the augmentation of the high-frequency components that deteriorate as a result of image enlargement. A codebook-based method, which utilizes the relationship of the low- and high-frequency image components, is a proven method of high-quality image enlargement. However, this method requires a significant calculation time to be employed for a codebook search process, and thus is not suitable for real-time processing. In this paper, we propose a fast image-enlargement algorithm that includes a hierarchical predefined codebook and a fast edge-patch detection rule. A hierarchical codebook search reduces the number of comparisons that need to be made in comparison with a conventional full codebook search. In addition, applying the augmentation of the high-frequency components to only the edge patches detected under the proposed rule reduces the calculation time. The experimental results show that the calculation time of the proposed method is around 180 times faster than that of the fastest existing codebook-based method. The proposed method realizes high-quality image enlargement in terms of both objective and subjective evaluations in comparison with conventional methods.*

Keywords: Image enlargement, Image scaling, Hierarchical codebook, Edge detection, Fast algorithm

1. **Introduction.** In recent years, high-resolution displays have become widely used in areas such as high-definition televisions, mobile devices, and smart phones. Fast high-quality image enlargement is a very important technology in displaying low-resolution images on such devices. Classical image-enlargement methods – such as nearest neighbor interpolation (NNI), bilinear interpolation (BLI), and cubic convolution interpolation (CCI) – are based on interpolation using kernels [1, 2]. These interpolation-based methods achieve fast smooth image-enlargement; however, they cannot restore the high-frequency image components lost in the sampling process, and therefore cannot preserve clearly the

step edges and peaks of an image. This is caused by the fact that the high-frequency image components beyond the Nyquist frequency cannot be restored using these simple kernel-based methods.

To address this problem, various advanced image-enlargement methods have been proposed. For example, there are directional methods [3, 4], multiple kernel-based methods [5], orthogonal transform-based methods [6, 7], a linear extrapolation method [8], neural network based methods [9, 10, 11], fuzzy-inference based methods [12, 13], a bivariate rational interpolation method [14], and area area-pixel-model-based interpolation methods [15, 16]. All these methods exhibit better performance than simple kernel-based methods; however, they are all complex – which makes practical implementation difficult – or are all based on the assumptions that are not always correct in practical situations – and thus, they may give poor results.

High-quality image enlargements are achieved by employing various methods to restore the high-frequency components that deteriorate as a result of image enlargement. Estimation methods using the given image itself are effective for this restoration [17, 18, 19]. We have proposed codebook-based methods to represent the relationship of the low- and high-frequency image components [18, 19, 20]. Self-decomposed-codebook (SDC)-based image enlargement is achieved by augmenting a low-resolved enlarged image with the high-frequency components estimated from a given image itself [18, 19]. The estimation of the high-frequency image components is performed by a codebook compiled by the decomposition of the given image. The SDC technique can, in fact, produce an enlargement that is satisfactory, but it suffers from high computational cost and a considerable memory storage requirement because a codebook needs to be generated from each given image and the size of codebook is quite large – approximately the same size as the number of pixels in the given image. Predefined-codebook (PDC)-based image enlargement, which is suitable for various images, enables the skipping of the generation process of a codebook [20], and the computational complexity of a PDC is less than of an SDC. However, it still requires significant calculation time for a codebook search process because a PDC contains many code words. For practical applications, a further reduction in the calculation time is definitely desired.

To resolve these difficulties, we propose the use of a hierarchal predefined codebook (HPDC), which reduces the calculation time for the codebook search process. We also propose a fast-edge patch detection rule that easily divides an input image into an edge areas and a non-edge areas. By applying the augmentation of the high-frequency components to only the edge areas, we further reduce the calculation time for the codebook search process. This rule can be applied to other high-frequency-component-enhancement-based methods [17-20]. The proposed algorithm achieves both fast calculation and high-quality image enlargement. To show the effectiveness of the proposed method, we compare the enlarged results of the HPDC with the conventional methods. The experimental results are evaluated by a mean-square-error by the required CPU time for image quality and the calculation time, respectively.

2. **Predefined-Codebook-Based Image-Enlargement Framework.**

An image can be represented as a composition of low- and high-frequency image components, as shown in Figure 1. Generally, the high-frequency image component is lost in the process of kernel-based image enlargement such as CCI. If the codebook-based method can estimate the high-frequency components from the low-frequency components, the enlarged image can be augmented with the estimated high-frequency components. Therefore, the codebook should represent the relationship of the low- and high-frequency components.

(a) Ideal image (b) Low-freq. component (c) High-freq. component

FIGURE 1. Relationship of low- and high-frequency components. Ideal image (a) can be represented as a composition of low- (b) and high- (c) frequency image components.

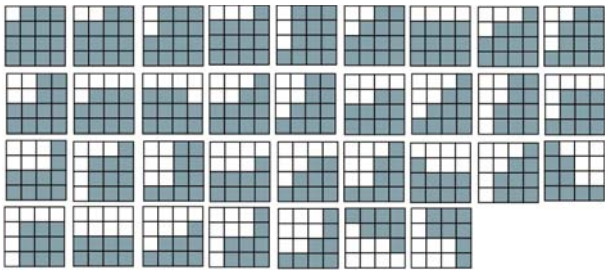

FIGURE 2. Fundamental edge patterns

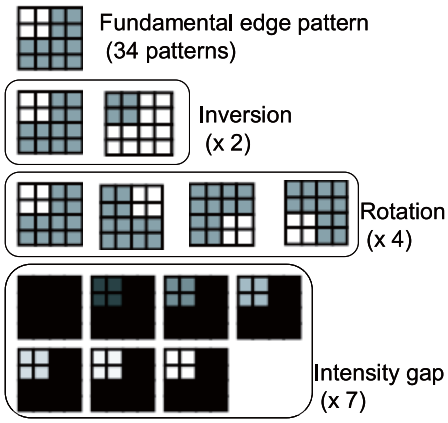

FIGURE 3. Translated variants for a fundamental edge pattern

In this section, we introduce a scheme for generating a PDC [20], and then extend it to the application of an HPDC codebook.

2.1. **Predefined codebook generation.** A PDC is generated from a fundamental edge patterns, as shown in Figure 2. The 34 small image patches in the figure represent a typical variation in intensities in a 4×4 pixels image patch. By considering rotations every 90 degrees, inversions, and seven different intensity gaps on the basis of the fundamental edge patterns, as shown in Figure 3, a universal PDC entailing 1,904 code words is obtained.

Figure 4 shows an overview of the generation of a PDC. To obtain the relationship of the low- and high-frequency image components, a low-pass filter is applied to a geometrically similar edge pattern (12×12 pixels) of the given fundamental pattern (4×4 pixels). The center of the low-pass filtered image (4×4 pixel patch) is selected as a low-frequency code word $\boldsymbol{x}_l$, and a corresponding high-frequency code word $\boldsymbol{x}_h$ is calculated by subtracting

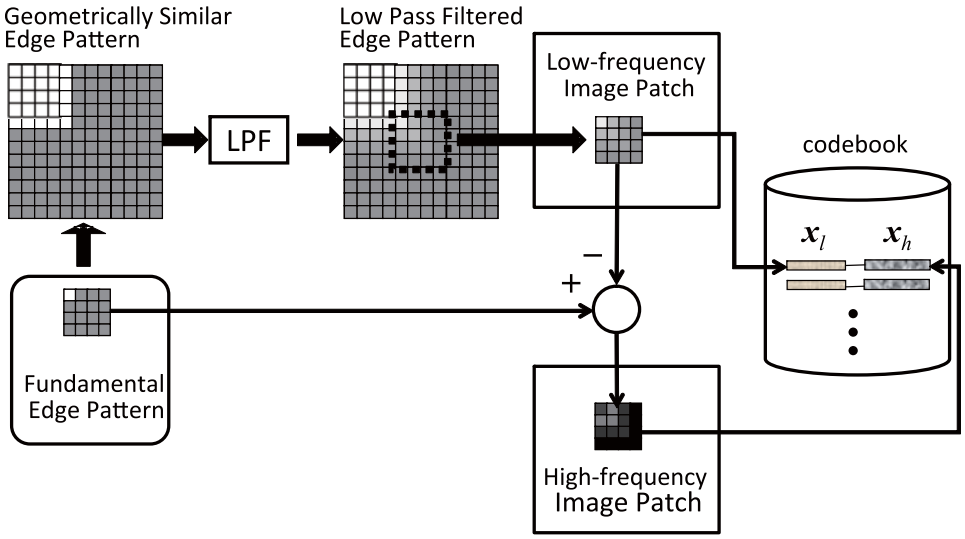

FIGURE 4. Generation of predefined codebook

the original patch from the low-pass filtered patch. After applying the above process to all the code words, pairs of $\{\boldsymbol{x}_h^p, \boldsymbol{x}_l^p\}$, $(p = 1, 2, \cdots, 1904)$ are obtained as the PDC.

2.2. **Codebook reference.** To estimate the high-frequency image components, a vector X_l^q (4×4 pixels) is generated as a low-frequency patch reference from the cubic-convolution-interpolated image to an arbitrary coordinate q. The vector X_l^q is compared with all patches x_l^p, and the best-matching patch is determined on the basis of a local distance measure. Therefore, we seek a pixel coordinate q for each X_l^q, which minimizes the following function:

$$p^* = \arg\min_p d(x_l^p, X_l^q), \tag{1}$$

where pixel coordinates p, q run over all points in their respective images. The function $d(\cdot, \cdot)$ is a local distance measure between two small patches. In this paper, the Euclidean distance is used as below:

$$d(x_l^p, X_l^q) = ||x_l^p - X_l^q||. \tag{2}$$

2.3. **Estimation of lost high-frequency components.** After the determination of the corresponding $x_h^{p^*}$ for all patches X_l^q through the codebook reference discussed above, all overlapped patches for a single pixel are handled as follows: In the case of a 4×4 patch size, the number of shared patches for a single pixel is 16 except for pixels near the edges of an image and in the corners. In these regions, a single pixel shares less patches. The corresponding pixel intensities in the shared patches are totaled and then divided by the overlapped number. Consequently, the high-frequency components of the enlarged image are created. The resultant high-frequency image is superimposed on the cubic-convolution-interpolated image to estimate the high-frequency components. Then we accomplish an image enlargement with perceptual sharpness.

3. **Hierarchical Predefined Codebook.** Since the PDC consists of 1,904 code words, 1,904 comparisons need to be performed for Equation (1), which requires a significant calculation time in a software implementation using a single CPU. In addition, from the viewpoint of special-purpose hardware implementation, 1,904 processing elements – needed to calculate Equation (1) – are required to realize a full parallel calculation, and this requires significant hardware resources, and therefore it is difficult to realize in current devices.

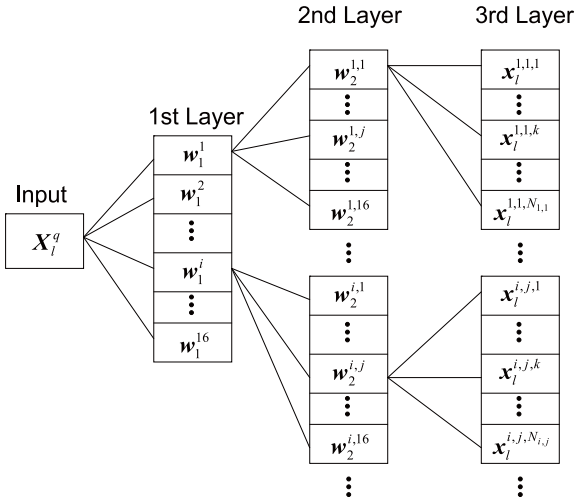

FIGURE 5. Hierarchal predefined codebook

To reduce the calculation time and the required hardware resources, we propose a hierarchical PDC for the augmentation of high-frequency component image enlargement.

3.1. **Hierarchical rearrangement of codebook.** We rearrange the PDC into a three-layered hierarchical structure. Figure 5 shows the structure of the HPDC. First, we use 16 reference vectors w_1^i, $(i = 1, 2, \cdots, 16)$ for the 1st layer. The position of each reference vector is decided by the vector quantization of the low-frequency code words $\boldsymbol{x}_l$. The vector quantization is performed using batch-learning neural gas [21, 22]. After reference vector allocation, the 1,904 code words are divided into 16 clusters in the 1st layer.

Next, we use another 16 reference vectors $w_2^{i,j}$, $(j = 1, 2, \cdots, 16)$ for each cluster i in the 2nd layer. Similarly, the vector quantization of divided code words in the 1st layer is performed to decide the positions of reference vectors. After the vector quantization of the 1st and 2nd layers, the 1,904 code words are divided into 256 clusters using reference vectors w_1^i and $w_2^{i,j}$. Each cluster consists of $N_{i,j}$ $(\Sigma N_{i,j} = 1,904)$ code words.

In our experiment, the maximum number of $N_{i,j}$ is 32, the minimum number of $N_{i,j}$ is 2, and the average number of $N_{i,j}$ is 7.43.

3.2. **Hierarchical codebook reference.** According to the hierarchical rearrangement of the codebook, Equation (1) is modified in line with the three steps below.

$$i^* = \arg\min_i d\left(w_1^i, X_l^q\right), \tag{3}$$

$$j^* = \arg\min_j d\left(w_2^{i^*j}, X_l^q\right), \tag{4}$$

$$k^* = \arg\min_k d\left(x_l^{i^*j^*k}, X_l^q\right). \tag{5}$$

Equations (3)-(5) take 16, 16, and average 7.43 times comparisons, respectively. The total number of average comparisons becomes 39.43. Therefore, a hierarchically codebook reference can be reduced to approximately 1/50th the size of a conventional codebook reference. Conversely, from the viewpoint of special-purpose hardware implementation, 16, 16 and 32 processing elements are required for the 1st, 2nd and 3rd layer of the hierarchically codebook reference, respectively. The 1st, 2nd and 3rd layers can be implemented as a pipelined architecture; therefore, only 64 processing elements are required to realize a full parallel calculation. The required hardware resources are likewise reduced.

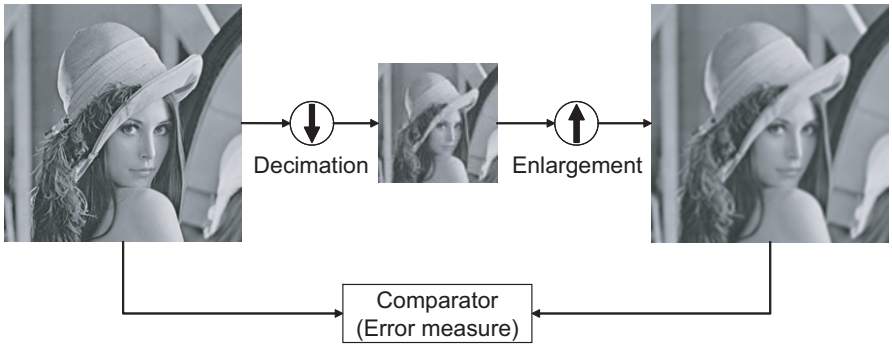

FIGURE 6. Error measure layout for objective evaluation

4. **Fast Edge Detection Rule.** As shown in Figure 1(c), a high-frequency image can be roughly divided into edge and smooth areas. If the input patch X_l^q is detected as a non-edge area, then the codebook search-and-augmentation process can be skipped. In this section, we propose a simple, fast-edge-detection rule to reduce the calculation time for image augmentation.

First, we calculate a normalized vector $\acute{X}_l^q$ by using the following function.

$$\acute{X}_l^q = X_l^q - \mu, \tag{6}$$

where μ is a mean of X_l^q; therefore, the input patch X_l^q is normalized to zero mean.

Next, an edge flag is calculated by the following function.

$$flag = \begin{cases} 1, & \max(\acute{X}_l^q) > \gamma \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

where the function max() selects the maximum vector element of $\acute{X}_l^q$, and γ is a threshold that determines whether the input patch X_l^q is an edge area. After the flag is decided, only edge patches that have $flag = 1$ are augmented by the HPDC.

5. **Experimental Results and Discussion.** To show the effectiveness and validity of the HPDC, we compared the enlarged results of the proposed method with conventional enlargement methods.

Error measures are used to objectively compare the enlarged image with the original one, as shown in Figure 6. First, the original image is decimated by a factor of 2 and then enlarged by a factor of 2. Next, the original and enlarged images are compared using an error measure. The decimation operation involves antialiasing filtering with the same parameter of LPF followed by downsampling. As an error measure, the mean-squared error (MSE) is used in this paper. The MSE is simply the mean of the squared differences for every channel for every pixel. By denoting x_{rc} as the value of pixel r in channel c of the original image, y_{rc} as the value of pixel r in channel c of the compared image, n as the number of pixels, and c as the number of channels, the MSE can be obtained by the following equation:

$$MSE = \frac{1}{nm} \sum_{r=1}^{n} \sum_{c=1}^{m} (x_{rc} - y_{rc})^2 . \tag{8}$$

In our experiments, we use 12 images (grayscale, 256×256 pixels) selected from the SIDBA database, which are royalty-free and have been used in other computer graphics performance tests, and are often referred to as "standard images".

5.1. **Results.** Six conventional methods are selected for comparison: BLI, advanced-area-pixel-model-based method (AAPM) [16], CCI [2], nonlinear extrapolation (NE) [17], SDC [18] and PDC-based enlargement [20]. BLI is a classical and simple interpolation method. AAPM is an area-pixel-model-based scaling algorithm that extends the Winscale method [15]. CCI and NE are selected as the most well-known interpolation and high-frequency-component-enhancement-based enlargement methods, respectively. SDC and PDC are codebook-based methods. The HPDC-based enlargement methods proposed in this paper (HPDC1 and HPDC2) are compared with these conventional methods.

HPDC1 augments all input patches in a manner similar to that for SDC and PDC. HPDC2 enhances only the edge patches, and the non-edge patches skip the codebook search process. We used $\gamma = 10$ for the threshold in the evaluation in Equation (7). Figure 7 shows the result of edge detection for "Lenna". The input low-frequency image component was divided into the edge and non-edge areas shown in Figures 7(a) and 7(b), respectively, by the proposed rule. In the case of "Lenna", 40% of input patches were selected as edge areas. After edge area selection, HPDC2 performed a hierarchical codebook search process for only the edge areas, and then generated the estimated high-frequency component shown in Figure 7(c). By the augmentation of the input low-frequency image with the estimated high-frequency image component, the augmented image was obtained as the enlarged result shown in Figure 7(d).

(a) (b) (c) (d)

FIGURE 7. Result of edge detection for Lenna: (a) edge area, (b) non-edge area, (c) estimated high-frequency components, (d) augmented image

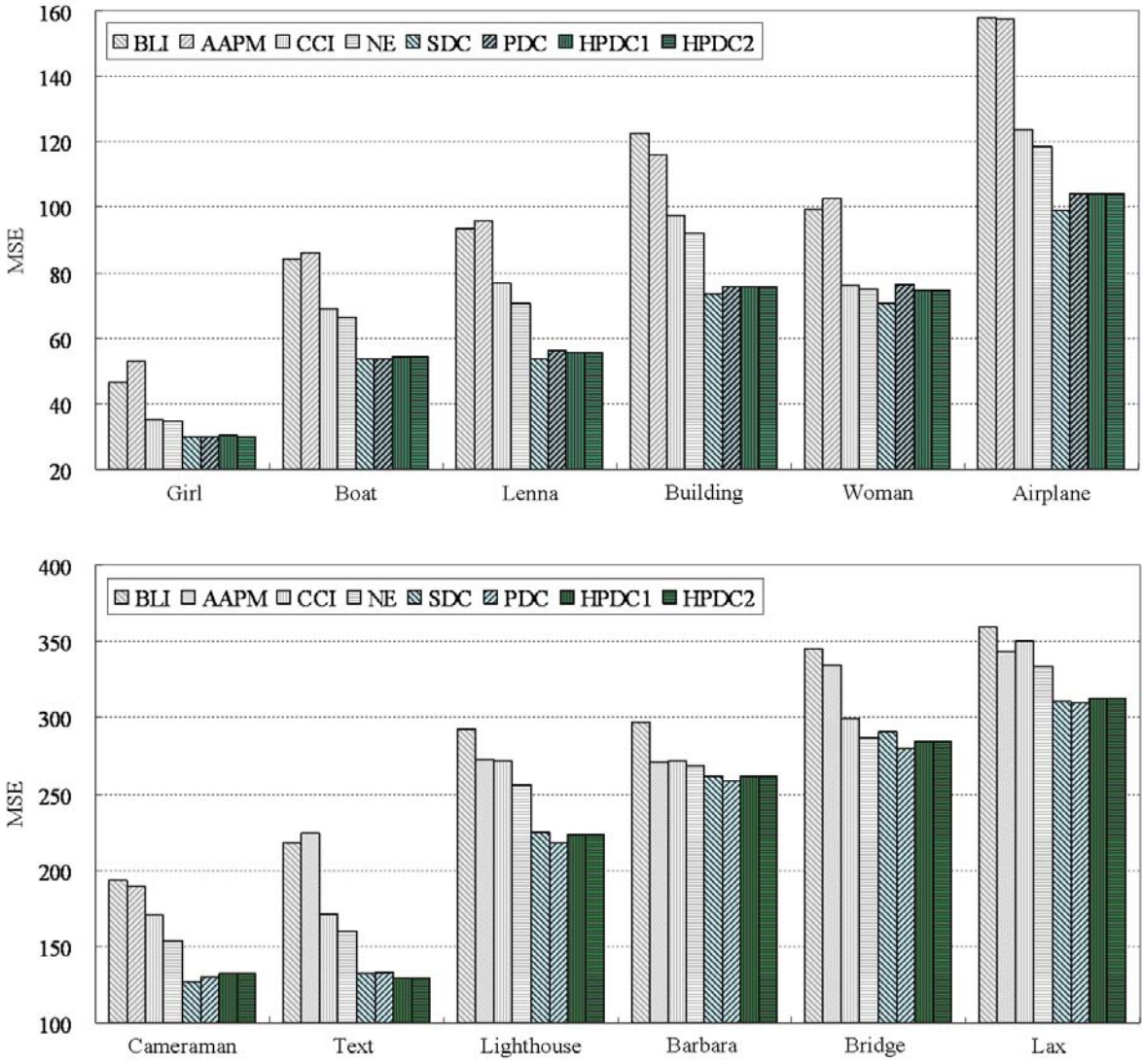

FIGURE 8. Comparison of MSE for image quality

TABLE 1. Evaluation effects of edge-area detection in HPDC2

	Ratio of non-edge area	MSE difference
Min (image)	27% (Bridge)	0.00 (Building and Lighthouse)
Average	52%	0.18
Max (image)	70% (Cameraman)	0.50 (Bridge)

5.1.1. *Objective evaluation.* MSE results for enlargement are shown in Figure 8 for grayscale images so as to perform an objective evaluation of image quality. As shown in the results, HPDC1 and HPDC2 were superior to conventional methods (BLI, AAPM, CCI and NE), and were on par with codebook-based methods (SDC and PDC). Consequently, we confirmed that the proposed methods yield equivalent performance with respect to MSE evaluations.

Table 1 shows the effects of proposed edge area detection rule on HPDC2. Average 52% of input patches were selected to the non-edge area such as Figure 7(b). Although these patches skipped the high-frequency components enhancement process, the MSE difference

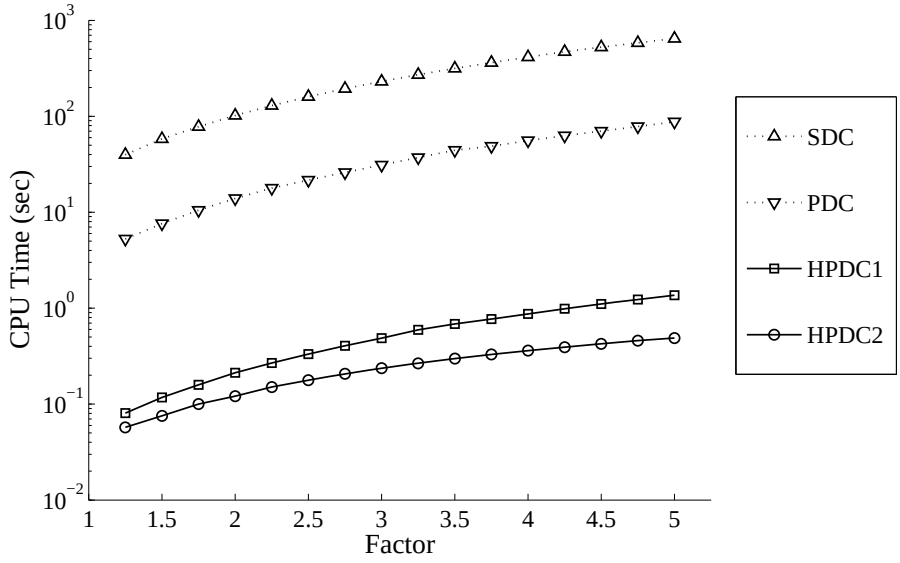

FIGURE 9. CPU time comparison of codebook based image enlargement methods

between HPDC1 and HPDC2 is quite small. The result shows that the proposed scheme is effective to reduce the calculation time without any effect on image quality.

To evaluate the calculation time of codebook-based enlargement methods, we measured CPU times of software implementation (Visual C++ compiler, single thread programming, Intel Core2Duo 3.16 GHz CPU, 4 GB RAM). The CPU Time required for enlargement is shown in Figure 9. The x-axis represents an enlargement factor from 1.25 to 5 at a 0.25 increment. For instance, the enlarged image is 640×640 pixels when the factor is 5. Each data point is calculated by averaging 12 enlarged images. When the factor is 5; SDC, PDC, HPDC1 and HPDC2 required 646.31, 87.49, 1.37 and 0.49 second, respectively. The results show that HPDC2 was approximately 1320 times faster than SDC and 180 times faster than PDC.

These results also show that the proposed HPDC2 achieved both fast calculation and high-quality image enlargement.

5.1.2. *Subjective evaluation.* To subjectively assess the performance of the enlargement methods, we show the enlarged results with respect to "Lenna" in Figure 10. In this figure, each image corresponds to a part of the original image cropped by 128 × 128 pixels. The images in Figure 10 from left to right and top to bottom are (a) ideal image and the enlarged results of (b) CCI, (c) NE, (d) SDC, (e) PDC and (f) HPDC2. As the figure shows, the CCI image is blurred. In contrast, the images by NE, SDC, PDC and HPDC2 are comparatively sharper. However, we observe some artifacts along the edge of the hat line in the result of NE. By comparing PDC and HPDC2, it is difficult to find any differences in the results. The subjective results also suggest that HPDC2 achieves high-quality image enlargement similar to PDC. Comparing SDC and HPDC2, we can observe that the result of SDC is smoother than that of HPDC2 in terms of smoothness of the edge of the hat. The image quality by SDC tends to be marginally better than that by HPDC2, but it takes considerable calculation time, as shown in Figure 9.

5.2. **Discussion.** The proposed HPDC2 achieved calculation faster than the existing fastest codebook-based method. However, the performance of CPU implementation is not sufficient for real-time applications such as for high-definition televisions (1,920×1,080

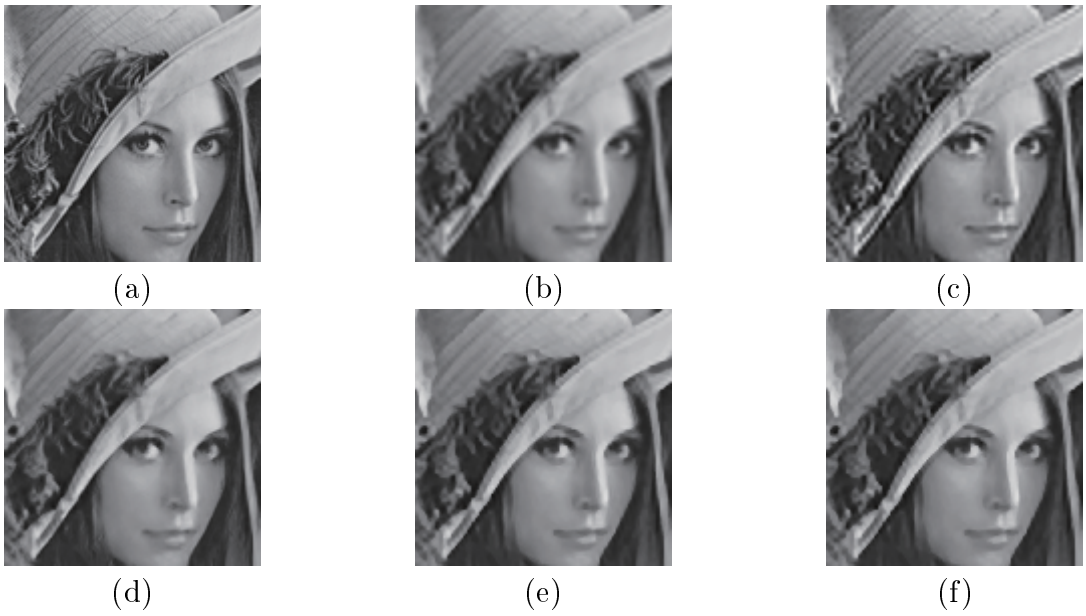

FIGURE 10. The original gray-scale image and the enlargement results: (a) Ideal, (b) CCI, (c) NE, (d) SDC, (e) PDC, (f) HPDC2 (proposed)

pixels, 60 frames per second). A solution for this problem is a digital hardware implementation that allows massive parallel processing. In this section, we discuss a possible HPDC hardware implementation.

To realize the HPDC into a field-programmable-gate-array (FPGA), it needs 548,352 bits of FPGA internal memory. In particular, each code word $\{x_l, x_h\}$ has 32 pixels, and these pixels are represented as a 9-bits accuracy in the HPDC; thus, the total memory is 548,352 bits [32 (pixel/word) $\times$ 9 (bit/pixel) $\times$ 1,904 (word)]. It is compact and easy to implement into the FPGA. For example, Xilinx xc3s4000, which is a low-cost FPGA built on CMOS 90-nm technology, has 1,769,472 bits of block RAM (FPGA internal memory) [23]. Similarly, Xilinx xc6vsx475t, which is a recent high-performance FPGA built on CMOS 40-nm state-of-the-art technology, has 39,223,296 bits of block RAM [23]. Therefore, the FPGA implementation of the proposed method is free from any memory limitation.

The process of Equations (3), (4), (5) and (2) is commonly called as "winner-take-all" (WTA). As WTA is widely used in fields such as image processing and neural networks, several digital hardware implementations have been proposed to realize high-speed processing [24, 25]. The proposed HPDC requires 64 processing elements, as mentioned in Section 3.2. A similar architecture [25], Xilinx xc2v6000, which is an obsolete FPGA built on CMOS 150-nm technology, can implement 300 processing elements for WTA processing. Therefore, the FPGA implementation of the proposed method is also free from any hardware resources limitation.

The proposed method has data parallelism similar to other image processing methods. For example, if the given image can be sliced into four divided images, four enlargement circuits can run in parallel. This means that the proposed method is suitable for massively parallel architecture, and the hardware implementation can realize real-time processing. According to these estimates, the proposed method can be implemented on standard FPGA boards.

6. **Conclusions.** In this paper, to resolve the difficulties in conventional codebook-based image enlargement methods, we proposed fast image-enlargement algorithms that include

an HPDC and a fast edge-patch detection rule. The proposed HPDC was described as a three-layered structure that realized fast calculation in the codebook search process. By applying the augmentation of the high-frequency component to only the edge areas detected by the proposed rule, the calculation time for the codebook search process was also reduced. Based on the experimental results, the proposed method realized fast calculation in comparison with other codebook-based enlargement methods, without image degradation.

The algorithm for the proposed method is not complex, and hence can be easily implement in digital hardware. The required hardware resource are fewer than other codebook-based methods. The proposed enlargement method can be processed in parallel because the estimations of the high-frequency image patches for every LPF image patch are calculated independently. Thus, the proposed method has potential for a wide range of image-enlargement applications.

In future study, the proposed method will be implemented on an FPGA board and applied to the real-time processing of moving images.

Acknowledgment. This study was supported by the Japan Society for the Promotion of Science under a Grant-in-Aid for Scientific Research C (No. 21500230) and a Grant-in-Aid for Scientific Research B (No. 24300092).

REFERENCES

[1] J. S. Lin, *Two-Dimensional Signal Processing and Image Processing*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1990.

[2] R. G. Keys, Cubic convolution interpolation for digital image processing, *IEEE Trans. Acoust. Speech Signal Process.*, vol.26, no.6, pp.1153-1160, 1981.

[3] K. Jensen and D. Anastassiou, Subpixel edge localization and the interpolation of still images, *IEEE Trans. Image Process.*, vol.4, no.3, pp.285-295, 1995.

[4] X. Li and M. T. Orchard, New edge-directed interpolation, *IEEE Trans. Image Process.*, vol.10, no.10, pp.1521-1527, 2001.

[5] A. M. Darwish and M. S. Bedair, An adaptive resampling algorithm for image zooming, *Proc. SPIE*, vol.2466, pp.131-144, 1996.

[6] S. A. Martucci, Image resizing in the discrete cosine transform domain, *Proc. of IEEE Int. Conf. Image Process.*, pp.2244-2247, 1995.

[7] S. G. Chang, Z. Cvetkovic and M. Vetterli, Resolution enhancement of images using wavelet transform extrema extrapolation, *Proc. of IEEE Int. Conf. Acoust. Speech Signal Process.*, vol.4, pp.2379-2382, 1995.

[8] T. Aso, N. Suetake and T. Yamakawa, A weighted linear extrapolation-based simple image enlargement algorithm, *Intelligent Automation and Soft Computing*, vol.12, no.3, pp.345-353, 2006.

[9] N. Plaziac, Image interpolation using neural networks, *IEEE Trans. Image Process.*, vol.8, no.11, pp.1647-1651, 1999.

[10] F. M. Candocia and J. C. Principe, Superresolution of images based on local correlations, *IEEE Trans. Neural Netw.*, vol.10, no.2, pp.372-380, 1999.

[11] F. Pan and L. Zhang, New image super-resolution scheme based on residual error restoration by neural networks, *Opt. Eng.*, vol.42, no.10, pp.3038-3046, 2003.

[12] A. Taguchi and T. Kimura, Edge-preserving interpolation by using the fuzzy inference, *Proc. of SPIE*, vol.4304, pp.98-105, 2001.

[13] S. Carrato, G. Ramponi and S. Marsi, A simple edge-sensitive image interpolation filter, *IEEE International Conf. on Image Processing*, pp.711-714, 1996.

[14] Y. F. Zhang, S. Gao, C. Zhang and J. Chi, Application of a bivariate rational interpolation in image zooming, *International Journal of Innovative Computing, Information and Control*, vol.5, no.11(B), pp.4299-4307, 2009.

[15] C. H. Kim, S. M. Seong, J. A. Lee and L. S. Kim, Winscale: An image-scaling algorithm using an area pixel model, *IEEE Trans. on Circuits and Systems for Video Tech.*, vol.13, no.6, pp.549-553, 2003.

[16] P.-Y. Chen and C.-Y. Lien, A novel image scaling algorithm based on area-pixel model, *International Journal of Innovative Computing, Information and Control*, vol.6, no.3(A), pp.1039-1048, 2010.
[17] H. Greenspan, C. H. Anderson and S. Akber, Image enhancement by nonlinear extrapolation in frequency space, *IEEE Trans. Image Process.*, vol.9, no.6, pp.1035-1048, 2000.
[18] N. Suetake, M. Sakano and E. Uchino, Image super-resolution based on local self-similarity, *Optical Review*, vol.15, no.1, pp.26-30, 2008.
[19] H. Kawano, N. Suetake, B. Cha and T. Aso, Sharpness preserving image enlargement by using self-decomposed codebook and Mahalanobis distance, *Image Vision Comput.*, vol.27, no.6, pp.684-693, 2009.
[20] H. Kawano, H. Tamukoh, N. Suetake, B. Cha and T. Aso, Image enlargement with high-frequency component augmentation based on predefined codebook describing edge blurring properties, *Optical Review*, vol.17, no.5, pp.447-453, 2010.
[21] T. Martinetz, S. Berkovich and K. Schulten, 'Neural gas' network for vector quantization and its application to time series prediction, *IEEE Trans. Neural Netw.*, vol.4, no.4, pp.558-569, 1993.
[22] B. Hammer, A. Hasenfus and T. Villmann, Magnification control for batch neural gas, *Neurocomputing*, vol.70, no.7-9, pp.1225-1234, 2007.
[23] Xilinx Inc., *http://www.xilinx.com/*.
[24] D. C. Hendry, Comparator trees for winner-take-all circuits, *Neurocomputing*, vol.62, pp.389-403, 2004.
[25] H. Tamukoh, K. Horio and T. Yamakawa, Fast learning algorithms for self-organizing map employing rough comparison WTA and its digital hardware implementation, *IEICE Trans. on Electronics*, vol.E87-C, no.11, pp.1787-1794, 2004.