BLOCK-ADAPTIVE INTERPOLATION FILTER FOR SUB-PIXEL MOTION COMPENSATION

Jaehyun Cho, Dong-Bok Lee, Shin Cheol Jeong, Byung Cheol Song

School of Electronic Engineering, Inha University
Yonghyun-dong, Nam-gu, 402-751, Incheon, Republic of Korea
phone: + (82) 32-860-7413, email: bcsong@inha.ac.kr

ABSTRACT

Adaptive interpolation filtering for sub-pel motion estimation is one of several superior techniques of ITU-T KTA CODEC to the H.264/AVC CODEC. However, the adaptive interpolation filtering has a limitation in coding efficiency because of its frame-based update strategy of filter coefficients. In order to overcome such a problem, this paper presents a block-adaptive interpolation filtering using learning-based super-resolution. The proposed block-adaptive interpolation filtering for quarter-pel motion estimation consists of two steps: two-times up-scaling of half-pel accuracy and subsequent two-times up-scaling of quarter-pel accuracy. The dictionary optimized for each step is employed to produce the precise up-scaled blocks. Simulation results show that the proposed algorithm improves coding efficiency up to 5.3% in comparison with the previous adaptive interpolation filtering for KTA.

1. INTRODUCTION

Recently, with rapid development of semiconductor and digital display, high definition (HD) video contents have been popular. To efficiently transmit or store such huge video data, high compression technology is required. For example, H.264/AVC [1] is the latest video coding standard to meet such requirement, which was jointly implemented by ITU-T (International Tele-communication Union) and MPEG (Moving Picture Expert Group). As a key compression tool for H.264/AVC, sub-pel motion compensation is composed of half-pel motion compensation using 6-tap filter of fixed coefficients, and subsequent quarter-pel motion compensation using bilinear interpolation. However, the fixed filter coefficients may often deteriorate coding efficiency because they never take into account spatial characteristics of every frame.

For a recent few years, VCEG (Video Coding Experts Group) of ITU-T has developed KTA (Key Technology Area) software as an interim process for next-generation video coding standard by evaluating a lot of new compression tools and adopting high performance tools among them. As one of dominant techniques for KTA, AIF (Adaptive Interpolation Filter) for sub-pel motion compensation [2-4] improved coding efficiency up to about 10% in comparison with its counterpart of H.264/AVC. However, the AIF does not sufficiently consider local characteristics in a frame because it updates filter coefficients on a frame basis. This is a major drawback of the AIF method. Recently, MPEG has launched new coding standard to replace H.264/AVC in the future, which is called HEVC (High Efficiency Video Coding) [5]. For sub-pel motion compensation, the up-to-date version of HEVC adaptively chooses one between a simple DCT (Discrete Cosine Transform)-based interpolation filter and directional interpolation filter. The interpolation method for HEVC is meaningful in terms of computational complexity, but it does not show better coding efficiency than the AIF for KTA.

On the other hand, so-called super-resolution (SR) algorithms [6-9] have been developed as the most promising up-scaling approach. A typical SR makes use of signal processing techniques to obtain a high resolution (HR) image (or a sequence) from multiple low-resolution (LR) images. In general, success of such SR schemes depends on existence of sub-pixel motion between adjacent LR images and accurate sub-pixel estimation. However, sub-pixel motion estimation among neighbor LR images requires not only huge computational cost, but also its accuracy is not guaranteed in certain environments. In order to solve the above-mentioned problem, a lot of single image-based SR methods such as learning-based SR algorithms have been devised [7-9]. In general, learning-based SR is composed of two phases: Off-line learning phase and on-line synthesis phase. At the learning phase, the training data, i.e., dictionary consisting of LR and HR patches is constructed. The LR and HR patch pairs are obtained from various training images. During the synthesis phase, the input LR image is super-resolved by using the dictionary. For each LR patch in the input image, its nearest neighbor LR patches are explored from the dictionary. The high frequency components of the input LR patch are synthesized using the best matched LR patches [9]. Since this learning-based SR provides superior visual quality to conventional FIR filters at the expense of large memory size, it can be an attractive solution to high performance interpolation.

In order to overcome the above-mentioned drawback of the previous AIF method, this paper proposes a block-adaptive interpolation filter (BAIF) using learning-based SR. The proposed algorithm improves the performance of sub-pel motion compensation by adaptively updating filter coefficients on a block basis without additional side information. The BAIF consists of two steps for half-pel interpolation and quarter-pel interpolation. In off-line learning phase, the optimal dictionary of each step is derived from various LR and HR training images. Simulation results show that the pro-

posed algorithm provides higher coding efficiency of up to 5.3% than the previous AIF for KTA.

2. PREVIOUS WORKS

This subsection describes several AIF methods for KTA. NSAIF (Non-Separable AIF) [2] interpolates a sub-pel with two-dimensional (2D) filter coefficients optimized at the pixel position. In Fig. 1, a one-dimensional (1D) 6-tap filter is applied to sub-pels such as *a, b, c, d, h,* and *l*. The samples *C1-C6* are used for the sub-pel positions *a, b, c*, and *A3-F3* for *d, h, l*. For each of the remaining sub-pel positions *e, f, g, i, j, k, m, n*, and *o*, the 6 × 6 filter coefficients are calculated. For all sub-pel positions, the optimal filter coefficients are calculated in a way that the prediction error energy is minimized. The filter coefficients can be updated on a frame basis.

Since SAIF (Separable AIF) [3] is based on two 1D filter coefficients, it can achieve light interpolation complexity without any penalty on coding efficiency in comparison with NSAIF. Note that the computational expense of the SAIF itself is reduced by 24% in case of 4 × 4 motion-compensated blocks.

As another low complexity AIF, DIF (Directional AIF) [4] employs a single 1D directional interpolation filter coefficients at each sub-pel location. The direction of the interpolation filter is determined according to the alignment of the corresponding sub-pixel with integer pixel samples. For example, two sub-pels *e* and *o* of Fig. 1 are interpolated by applying a 6-tap filter to six integer samples *A1, B2, C3, D4, E5*, and *F6*. Since all sub-pels are obtained using only 1D filter operations, the complexity of the DIF is significantly less than its counterparts. In the worst case, the interpolation complexity of the DIF is 1/3 of NSAIF and less than 1/2 of SAIF.

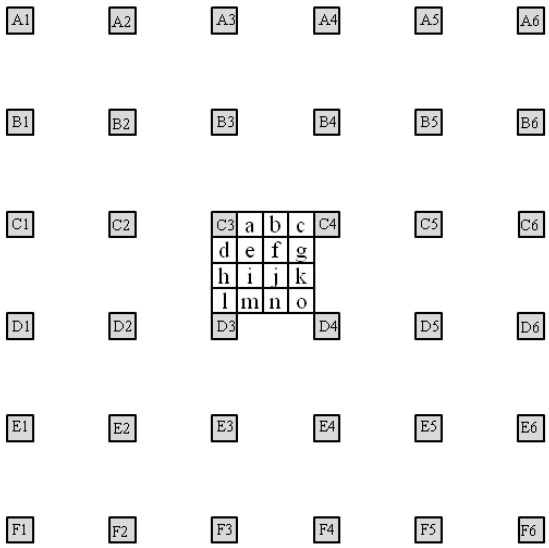


Fig. 1. Integer samples (shaded blocks with upper-case letters) and fractional sample positions (white blocks with lower-case letters).

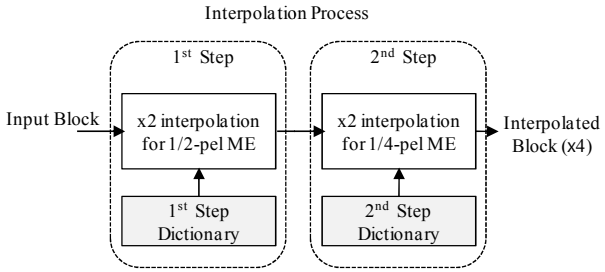


Fig. 2. The proposed interpolation for sub-pel motion estimation.

3. THE PROPOSED ALGORITHM

In order to overcome the drawback of the previous picture-adaptive AIF methods for KTA, we propose a block-adaptive AIF using learning-based SR as an interpolation tool for sub-pel motion estimation (see Fig. 2). At the first step, *b*, *h*, and *j* of half-pel accuracy are interpolated using the 1st step dictionary, and the remaining sub-pels, i.e., *a, c, d, e, f, g, i, k, l, m, n*, and *o* of 1/4-pel accuracy are interpolated using the 2nd step dictionary. From this dictionary-driven interpolation, we can achieve quarter-pel motion estimation and compensation guaranteeing high coding efficiency.

Like conventional learning-based SR algorithms [7-9], the optimal dictionaries can be derived from so-called learning phase, which is described in the following subsection.

3.1 Off-line Learning Phase

Fig. 3 describes the process to produce training images to the 1st and 2nd step dictionaries in off-line learning phase. As in Fig. 3, the training images of half-size resolution and quarter-size resolution are produced from the original HR images. Here, a well-known 5 × 5 Gaussian kernel is employed as an anti-aliasing filter. The 1st step dictionary for half-pel motion compensation is derived from half-size and quarter-size images. The 2nd step dictionary is generated from the HR images and the corresponding LR images which were reconstructed from the quarter-size images synthesized by the proposed SR based on the 1st step dictionary.

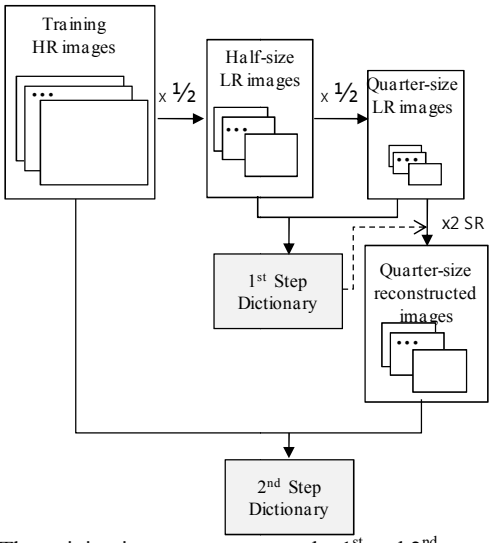


Fig. 3. The training images to generate the 1st and 2nd step dictionaries.

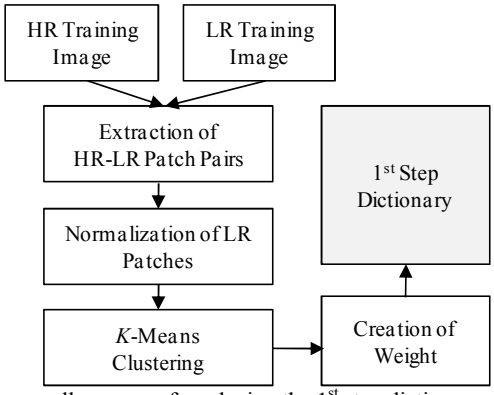


Fig. 4. The overall process of producing the 1st step dictionary.

Fig. 4 shows the overall process of producing the 1st step dictionary in more detail. First, all possible LR and HR patch pairs of $M \times M$ size are extracted from several quarter-size (LR) and half-size (HR) images. Let P_L^i and P_H^i denote the i-th LR and HR patches at the same spatial position. Fig. 5 describes an example of LR and HR patches when their sizes are set to 5 × 5. In this figure, all rectangles indicate HR pixels and grey rectangles indicate LR pixels. Each LR patch is extracted via proper overlapping with adjacent LR patches. In the current study, the $M/2$ pixels are overlapped between neighbor patches in both directions.

An input LR patch should be compared with candidate LR patches in the dictionary, and its HR patch is synthesized using the high frequency information corresponding to the candidate LR patch(es) with minimum distance. In order to improve the accuracy of such matching in the synthesis phase, Laplacian of LR patch is employed [9]. The Laplacian of each LR patch is produced by applying a 3 × 3 Laplacian operator to every pixel in the LR patch. Subsequently, Laplacian patches are normalized for further reliable matching. Let Q_L^i denote the normalized Laplacian of P_L^i.

Conventional learning-based SR requires as many patch pairs as possible to maintain reliable performance, which causes a tremendous memory cost as well as a significant matching computation. Therefore, we cluster similar LR and HR patch pairs. We apply K-means clustering based on Q_L to all patch pairs.

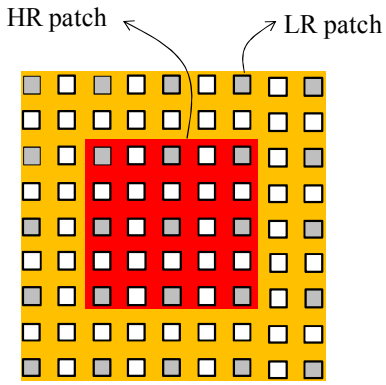


Fig. 5. An example of LR and HR patch pair.

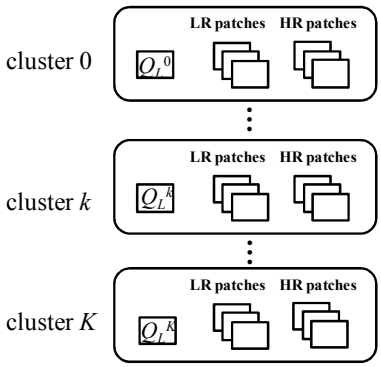


Fig. 6. The clustering results.

As a result, K Q_L cluster centers are obtained, and each cluster is indexed by its cluster center. Note that K is significantly smaller than the number of entire patch pairs extracted from LR and HR training images. Fig. 6 shows the clustering results. Let $P_L^{k,j}$ and $P_H^{k,j}$ be the j-th LR and HR patches in the k-th cluster. Then, $P_H^{k,j}$ can be computed from $P_L^{k,j}$ by the following equation:

$$P_H^{k,j}(s,t) = \sum_{u=0}^{M-1}\sum_{v=0}^{M-1} w_{st}^{k,j}(u,v) P_L^{k,j}(u,v), \qquad (1)$$

where (u, v) and (s, t) denote the pixel positions in the LR and HR patches, respectively. Now, we derive a common weight set $W^{(1)k}$, i.e., $\{w_{st}^{(1)k}(u,v) \mid 0 \le s,t,u,v \le M-1\}$ such that the squared sum of interpolation error by Eq. (1) is minimized for all LR and HR patches in the k-th cluster. In order to seek such an optimal weight set for each cluster, we employ popular LMS algorithm [10]. The superscript (1) of $W^{(1)k}$ indicates the 1st step. Finally, we can obtain the optimal 1st step dictionary $\{(Q_L^{(1)k}, W^{(1)k}) \mid 1 \le k \le K\}$.

The 2nd step dictionary is constructed in the same way as the 1st step dictionary. The only difference is that the 2nd step dictionary is trained from the original HR images and the half-size reconstructed images which are up-scaled from their corresponding quarter-size LR images using the 1st step dictionary as in the following subsection.

3.2 On-the-fly Interpolation Phase

For sub-pel motion estimation of each input block, two-step interpolation should be performed on an $M \times M$ block basis by using the 1st and 2nd step dictionaries as in Fig. 2. The 1st step interpolation is described in detail as follows:

Fig. 7 describes the 1st step interpolation process for half-pel motion estimation of an arbitrary 4 × 4 block. In this figure, the red-line rectangles indicate the integer-pixels of the motion-compensated 4 × 4 block. Prior to sub-pel motion estimation of the current 4 × 4 block, the half-pels in the blue region should be interpolated. In order to interpolate such half-pels, nine 5 × 5 LR patches are super-resolved in zigzag scan with overlapping of 3 LR pixels in both directions. Note that the half-pels pixels only in the blue region need to be synthesized. Half-pel motion estimation for the other size blocks can be operated similarly.

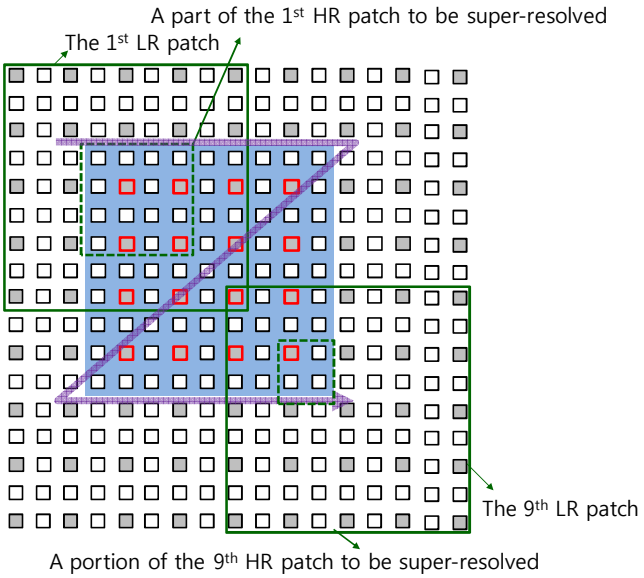


Fig. 7. The proposed half-pel interpolation process. Here, ■ and □ indicate integer- and half-pels, respectively.

The synthesis process of the 1st LR patch in Fig. 7 is depicted as follows. The normalized Laplacian Q_L^{in} for the input LR 5 × 5 block P_L^{in} is first derived. Then, the nearest candidate LR patch to Q_L^{in} is searched in the 1st step dictionary. In the current study, the sum of squared errors (SSE) is employed as the distortion measure for matching of Laplacian LR patches. Let $W_{best}^{(1)k}$ be the weight set corresponding to the best-matched Laplacian LR patch. From the input LR patch, we can produce the interesting half-pels of the dotted box inside the 5 × 5 HR patch (see Fig. 7) by using $W_{best}^{(1)k}$ and Eq. (1). Similarly, the remaining half-pels can be interpolated. For the half-pel positions in the overlapping region, multiple HR pixel values synthesized by Eq. (1) are averaged. At the same fashion, the quarter-pels can be derived from the integer- and interpolated half-pels by using the 2nd step dictionary. Note that the proposed algorithm does not have to transmit any side information related to filter coefficients to the decoder because the exact filter coefficients of every block can be obtained from the dictionaries in the decoder.

4. EXPERIMENTAL RESULTS

In order to evaluate the proposed algorithm, ten 1920 × 1080 video sequences of Table 1 are used. Also, six 3840 × 2160 training video sequences, which are not included in the test set, are employed to derive the 1st and 2nd step dictionaries.

The proposed interpolation algorithm was implemented on H.264 KTA software called JM 14.0 KTA2.6. For this experiment, RD optimization mode was off, CABAC was adopted for entropy coding, and the GOP structure was set to IPPPP. The first 5 frames of each test video sequence were encoded for 4 quantization parameters (QP), i.e., 22, 27, 32, and 37 in high profile.

Table 1. The comparison in terms of BD_rate (%).

	NSAIF	BAIF
Parkjoy	-5.32%	-9.22%
Parkscene	-6.01%	-9.53%
Crowdrun	-8.11%	-13.41%
Bluesky	-3.01%	-4.61%
Rolling tomatoes	-2.03%	-3.33%
Basketdrive	-6.58%	-9.32%
Rushhour	-6.31%	-10.34%
Traffic	-7.41%	-10.91%
BQTerrace	-7.56%	-11.30%
Station	-5.43%	-8.63%

The search range of integer-pel motion estimation was set to ±32. The size of LR patch was 5 × 5, and the number of clusters K was 512 for both 1st and 2nd step dictionaries.

Table 1 compares the proposed BAIF with the conventional single-pass NSAIF of KTA and 6-tap filter of H.264/AVC. They were compared in terms of averaged BD_rate. The fixed 6-tap filter of H.264/AVC was selected as a baseline to compute the BD-rates.

For example, the BAIF provides higher BD-rate of 5.3% at maximum than the conventional NSAIF for *Crowdrun* sequence. In general, the proposed algorithm shows much better coding efficiency for video sequences with complex textures or edges such as *Crowdrun* than homogeneous video sequences such as *Rolling tomatoes*. This is because the learning-based SR is normally very useful to accurately synthesize textures or edges.

In addition, Fig. 8 compares the proposed algorithm with fixed 6-tap filter of H.264 and AIF of KTA in terms of RD (Rate-Distortion) curves.

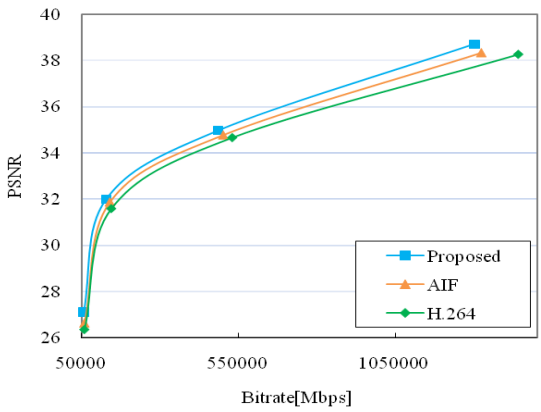
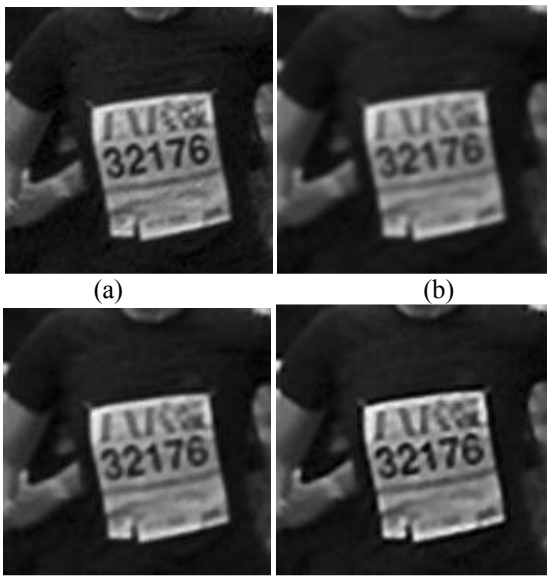


Fig. 8. RD curves of several algorithms for *Crowdrun* sequence.



(a) (b)

(c) (d)

Fig. 9. A part of the 5th frame of *Crowdrun*. (a) Original (b) H.264 (PSNR: 35.92dB, QP: 40) (c) NSAIF (PSNR: 36.13dB, QP: 37) (d) BAIF (PSNR: 36.41dB, QP: 35).

We can observe that the BAIF shows better coding performance in higher bit-rates. Also, Fig. 9 compares the proposed algorithm with previous works in terms of subjective visual quality. The *Crowdrun* sequence was encoded with proper QP values so that all the algorithms have almost same bit-rates, and then a part of the 5th decoded frame was chosen for comparison. We can see that the BAIF shows much better visual quality than the existing algorithms.

5. CONCLUSION

This paper presented a block-adaptive interpolation filtering which shows better RD performance as well as higher subjective visual quality than the conventional AIF for sub-pel motion estimation. The proposed algorithm employed the learning-based SR to maximize the interpolation accuracy. Also, the proposed algorithm does not have to transmit any side information related to filter coefficients because the exact filter coefficients of every block can be derived from the equivalent dictionaries in the decoder. Simulation results show that the proposed algorithm provides higher BD_rate of 13.4% at maximum than the conventional FIR filter of H.264/AVC.

ACKNOWLEDGMENT

This research was financially supported by the Ministry of Knowledge Economy (MKE) and the Korea Institute for Advancement of Technology (KIAT) through the Human Resource Training Project for Strategic Technology, and was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2010-0015861).

REFERENCES

[1] ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), Advanced Video Coding for Generic Audiovisual Service, May 2003.

[2] Y. Vatis, B. Edler, D. T. Nguyen, and J. Ostermann, "Two-dimensional nonseparable adaptive Wiener interpolation filter for H.264/AVC," ITU-T SG16/Q.6 Doc. VCEG-Z17, Busan, Korea, April 2005.

[3] S. Wittmann and T. Wedi, "Separable adaptive interpolation filter," ITU-T SG16/Q.6 Doc. C219, Geneva, Switzerland, June 2007.

[4] D. Rusanovskyy, K. Ugur, and J. Lainema, "Adaptive interpolation with directional filters," ITU-T SG16/Q.6 Doc. VCEG-AG21, Shenzhen, China, Oct. 2007.

[5] ITU-T Output Document, "Report of subjective test results of responses to the joint call for proposals (CfP) on video coding technology for high efficiency video coding (HEVC)," ITU-T SG16 Doc. JCTVC-A204, Dresden, Denmark, April, 2010.

[6] M. Zhao, M. Bosma, and G. de Haan, "Making the best of legacy video on modern displays," *J. Soc. Inf. Display*, vol. 15, no. 1, pp. 49-60, 2007.

[7] W. Freeman, T. Jones, and E. Pasztor, "Example-based super-resolution," *IEEE Computer Graphics and Applications*, vol. 40, no. 1, pp. 23-47, 2000.

[8] W. Fan and D. Yeung, "Image hallucination using neighbor embedding over visual primitive manifolds," *IEEE Proc. CVPR*, 2007.

[9] S. C. Jeong and B. C. Song, "Noise-robust super-resolution based on a classified dictionary," *Journal of Electronic Imaging*, Dec. 2010.

[10] S. Haykin, Adaptive Filter Theory, Chap. 9, 3rd ed., Prentice Hall, 1996.