DATA PRUNING-BASED COMPRESSION USING HIGH ORDER EDGE-DIRECTED INTERPOLATION

Dũng Trung Võ [1*], *Joel Solé* [2], *Peng Yin* [2], *Cristina Gomila* [2], *Truong Nguyen* [1]

[1] Video Processing Laboratory, UC San Diego. 9300 Gilman Drive, La Jolla, CA 92092.
[2] Thomson Corporate Research. 2 Independence Way, Princeton, NJ 08540.

ABSTRACT

This paper proposes a data pruning-based compression scheme to improve the rate-distortion relation of compressed images and video sequences. The original frames are pruned to a smaller size before compression. After decoding, they are interpolated to their original size by an edge-directed interpolation. The data pruning is optimized to obtain the minimal distortion in the interpolation phase. Furthermore, a novel high order interpolation is proposed to adapt the interpolation to many edge directions. This high order filtering uses extra surrounding pixels and achieves more robust edge-directed image interpolation. Simulation results are shown for both image interpolation and coding applications.

Index Terms— interpolation, data pruning, compression, spatial filtering, edge-directed interpolation, video coding.

1. INTRODUCTION

Nowadays, the request for higher quality video is emerging very fast. Video tends to higher resolution, higher frame-rate and higher bit-depth. New technologies to further reduce bit-rate are strongly demanded to combat the bit-rate increase of this high definition video, especially to meet the network and communication transmission constraints. In video coding, there are two main directions to reduce compression bit-rate. One is to improve the compression technology and the other one is to perform a preprocessing before compression.

The first direction can be seen from the development of the MPEG video coding standard, from MPEG-1 to H.264/MPEG-4 AVC. For most video coding standards, increasing quantization step size is used to reduce bit-rate [1]. However, this technique can result in blocky artifacts and other coding artifacts due to the loss of high frequency details. In the second direction, common techniques are low-pass filtering or downsampling (which can be seen as a filtering process) followed by reconstructing or upsampling at the decoder. For example, low-pass filters were adaptively used based on Human Visual System to eliminate high frequency information in [2] or to simplify the contextual information in [3]. Also, to reduce the bit-rate, some digital television systems uniformly downsized the original sequence and upsized it after decoding. The reconstructed video applying these techniques looked blur because they were designed to eliminate high-frequency information with the anti-aliasing filter before downsizing or with the low-pass filter in the preprocessing step.

[*] This work is done while Dũng Trung Võ was with Thomson Corporate Research.

This paper proposes a novel data pruning-based compression scheme to reduce the bit-rate while still keeping a high quality reconstructed frame. The original frames are first optimally pruned to a smaller size by adaptively dropping rows or columns prior to encoding. At the final stage, an interpolation phase is implemented to reconstruct the decoded frames to their original size. By avoiding filtering the remaining rows and columns, the reconstructed frames can achieve high quality from a lower bit-rate.

Main applications of interpolation are upsampling, demosaicking and displaying in different video formats. A wide range of interpolation methods has been discussed, starting from conventional bilinear and bicubic interpolations to sophisticated iterative interpolations such as projection onto convex sets (POCS) [4] and nonconvex nonlinear partial differential equations [5]. Another group of interpolation algorithms predicted the fine structure of the high resolution (HR) image from its low resolution (LR) version using different kinds of transform such as wavelet [6] or contourlet transform [7]. To avoid the jerkiness artifacts occurring along edges, edge-oriented interpolation methods were performed using Markov random field [8] or the LR image covariance [9].

All the above methods are for upsampling the same ratio in both horizontal and vertical directions. However, when interpolation is used along with data pruning, the method needs to adapt to the way of pruning the data and to the structure of surrounding pixels. For instance, there are pruning cases in which only rows or only columns are dropped and upsampling in only one direction is required. This paper develops a high order edge-directed interpolation scheme to deal with these cases. Another algorithm is also considered for the cases of dropping both rows and columns. The paper is organized as follows. Section II introduces the data pruning-based compression method and derives an optimal data pruning algorithm. Section III describes the high order edge-directed interpolation method. Results for interpolation and coding applications are presented in Section IV. Finally, Section V gives the concluding remarks.

2. OPTIMAL DATA PRUNING

The block diagram of the data pruning-based compression is shown in Fig. 1. Assume that the original frame I of size $M \times N$ is pruned to frame P of smaller size $(M - M_p) \times (N - N_p)$, where M_p and N_p are the number of dropped rows and columns, respectively. The purpose of data pruning is to reduce the number of bits representing the stored or compressed frame P'. Then, I' of the original size is obtained by interpolating P'.

In this paper, only the even rows and columns may be discarded, while the odd rows and columns are always kept for later interpolation. The block diagram of the data pruning phase is shown in Fig. 2.

Fig. 1. Block diagram of the data pruned-based compression.

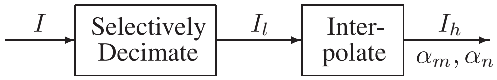


Fig. 2. Block diagram of the data pruning phase.

To simplify the analysis, the compression stage in Fig. 1 is ignored. In this phase, the original frame I is selectively decimated to the LR frame I_l for cases of dropping all even rows, all even columns and all even rows and columns. Then, I_l is interpolated to the HR frame I_h based on remaining odd rows and/ or columns after the selectively decimate phase. Finally, I_h is compared to I to decide which even rows and columns are dropped before compression. The mean squared error (MSE) between I_h and I is defined as

$$MSE = \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\left(I(m,n) - I_h(m,n)\right)^2 \tag{1}$$

Given a target MSE_{max}, the data pruning is optimized to discard the maximum number of pixels while keeping the overall MSE of I_h having dropped M_p rows and N_p columns less than MSE_{max}

$$\begin{aligned}(M_p, N_p)_{opt} = \arg \max_{M_p, N_p} \quad & \left(M_p N + N_p M - M_p N_p\right) \\ \text{s.t.} \quad & MSE(M_p, N_p) \leq MSE_{max}\end{aligned} \tag{2}$$

The location of the dropped rows and columns is indicated by α_m and α_n, respectively. If the k^{th} even column is dropped, then $\alpha_n(k) = 1$, otherwise $\alpha_n(k) = 0$. The similar algorithm is applied to rows. These indicators are stored as side information in the coded bitstream and are used for reconstructing the decoded frame. The line mean square error ($LMSE$) for one dropped column is defined as $\frac{1}{M}\sum_{m=1}^{M}\left(I(m,n) - I'(m,n)\right)^2$ and similarly for rows. From (2), lines with smaller $LMSE$ have higher priority to be dropped than lines with higher $LMSE$. Assume that the M_p rows and N_p columns with smallest $LMSE$ are dropped and the maximum $LMSE$ of these lines is $LMSE_{max}$. Then, the overall MSE in (1) becomes the averaged MSE of all dropped pixels

$$\begin{aligned}MSE(M_p, N_p) =& \frac{1}{N}\sum_{n=1}^{N}\alpha_n\left(\frac{1}{M}\sum_{m=1}^{M}\left(I(m,n) - I_h(m,n)\right)^2\right) \\ &+ \frac{1}{M}\sum_{m=1}^{M}\alpha_m\frac{1}{N}\left(\sum_{n=1}^{N}(1-\alpha_n)\left(I(m,n) - I_h(m,n)\right)^2\right) \\ \leq& \frac{1}{N}\sum_{n=1}^{N}\alpha_n LMSE_{max} + \frac{1}{M}\sum_{m=1}^{M}\alpha_m\left(\frac{N-N_p}{N}\right)LMSE_{max} \\ \leq& \left(1 - \left(1 - \frac{M_p}{M}\right)\left(1 - \frac{N_p}{N}\right)\right)LMSE_{max}\end{aligned} \tag{3}$$

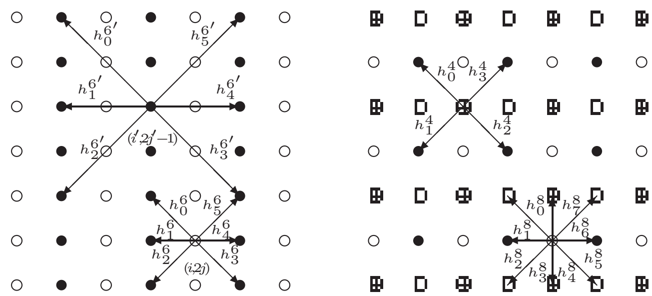
Therefore, the condition in (2) can be tightened to

$$\left(1 - \left(1 - \frac{M_p}{M}\right)\left(1 - \frac{N_p}{N}\right)\right)LMSE_{max} \leq MSE_{max} = \frac{255^2}{10^{\frac{PSNR_{min}}{10}}} \tag{4}$$

where $PSNR_{min}$ is the target minimal $PSNR$ that the reconstructed frame has to achieve. An example of the proposed optimal data pruning is shown in Fig. 3 for the 1^{st} frame of the sequence Akiyo. In Fig. 3a, the white lines indicate the dropped lines. With target $PSNR_{min} = 50\ dB$, the frame size is reduced from 720×480 to 464×320. The data pruned frame in Fig. 3b is more compact and requires smaller compressed bitstream than the original frame. Most dropped lines locate in flat areas where aliasing does not happen.

(a) Lines indicated for pruning (b) Pruned frame

Fig. 3. Data pruning for the 1^{st} frame of Akiyo sequence.



(a) $NEDI–6$ (b) $NEDI–8$

Fig. 4. Model parameters of high order edge-directed interpolation.

3. HIGH ORDER EDGE-DIRECTED INTERPOLATION

This section proposes a high order edge-directed interpolation method to upsize the downsized frames I_l in Fig. 1 and the data pruned frames P' in Fig. 2. In [9], the fourth-order new edge-directed interpolation (NEDI-4) is used to upsize only for the 2×2 ratio. This interpolation can only orient to edges in 2 directions and causes some artifacts in the intersections of more than 2 edges. The proposed method is a higher order interpolation that can adapt to more edge directions. The sixth-order edge-directed interpolation and eighth-order interpolation are developed for upsizing the cases with ratio 1×2 (dropping only rows or only columns) and ratio 2×2 (dropping both rows and columns), respectively.

3.1. Sixth-order Edge-directed Interpolation (NEDI-6)

The algorithm for dropping only columns is presented here. A similar algorithm is applied for the case of dropping only rows. The block diagram of NEDI-6 is shown in Fig. 5. First, P' is expanded to P'' of size $M \times N$ by inserting columns of zeros at the k^{th} column of P' if the column indicator $\alpha_n(k) = 1$. P'' is downsampled by 1×2 ratio to form I'_l of size $M \times \frac{N}{2}$. Then, columns of P'_l are mapped to the odd columns of the HR frame P'_h of size $M \times N$ by $P'_h(i, 2j-1) = P'_l(i,j)$. The even columns of P'_h are interpolated from the odd columns by a sixth-order interpolation

$$\hat{P}'_h(i,2j) = \sum_{k=-1}^{1}\sum_{l=0}^{1}h^6_{3l+k+1}P'_h(i+k, 2j+2l-1) = \boldsymbol{h^6}P'_{h_6} \tag{5}$$

where $\boldsymbol{h^6}$ is the vector of sixth-order model parameters and P'_{h_6} is the vector of 6-neighboring pixels of $P'_h(i, 2j)$ as shown in Fig. 4a. Assuming that $\boldsymbol{h^6}$ is nearly constant in a local window W, the optimal $\boldsymbol{h^6}$ minimizing the MSE between the interpolated $\hat{P}'_h(i, 2j)$ and original pixels in W can be calculated by

$$\boldsymbol{h^6}_{opt} = arg \min_{\boldsymbol{h^6}} \sum_{(i,2j) \in W}\left(P'_h(i, 2j) - \boldsymbol{h^6}P'_{h_6}\right)^2. \tag{6}$$

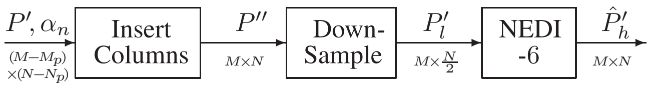


Fig. 5. Block diagram of the NEDI-6.

The geometric duality assumption states that the model vector $\boldsymbol{h^6}$ can be considered constant for different scales and so, it can be estimated from the LR pixels by

$$\boldsymbol{h^6}_{opt} = \boldsymbol{h^{6'}}_{opt} = arg\min_{\boldsymbol{h^{6'}}} \sum_{(i',2j'-1)\in W} \left(P'_h(i', 2j'-1) - \boldsymbol{h^{6'}} P'_{l_6}\right)^2 \quad (7)$$

where P'_{l_6} are 6-neighboring LR pixels of $P'_h(i', 2j'-1)$ and $\boldsymbol{h^{6'}}$ is the LR model parameter vector, as shown in Fig. 4a. $\boldsymbol{h^{6'}}$ is the edge-directed weight at LR scale and is applied to HR scale for interpolation. The optimal minimum MSE linear $\boldsymbol{h}$ is then obtained by

$$\boldsymbol{h^6}_{opt} = \left((A^6)^T A^6\right)^{-1} (A^6)^T \boldsymbol{y} \quad (8)$$

where $\boldsymbol{y}$ is the vector of all K mapped LR pixels $P'_h(i', 2j'-1)$ in W and A^6 is a $6 \times K$ matrix. The elements of the k^{th} column of A^6 are the 6-neighboring pixels of LR pixels. Finally, the column indicator α_m determines whether the columns in the final reconstructed frame are selected from the interpolated or from the data prunted frame

$$I'(m,n) = \begin{cases} P''(m,n) & \text{if } \alpha_n(n) = 0 \\ \hat{P}'_h(m,n) & \text{otherwise.} \end{cases}$$

3.2. Eighth-order Edge-directed Interpolation (NEDI-8)

This section develops an algorithm to deal with upsampling ratio 2×2. Similar to NEDI-6, the pixels in P' corresponding to the LR pixels downsampling by 2×2 in I are extracted to form the LR frame P'_L of size $\frac{M}{2} \times \frac{N}{2}$. The interpolation is performed using NEDI-4 as in [9] for the first round and NEDI-8 for the second round. Using the quincunx sublattice, 2 passes are performed in the first round. In the first pass, NEDI-4 is used to interpolate type 1 pixels (squares with lines in Fig. 4b) from the LR pixels (solid circles). In the second pass, type 2 pixels (squares) and type 3 pixels (circles) are interpolated from type 1 pixels and LR pixels.

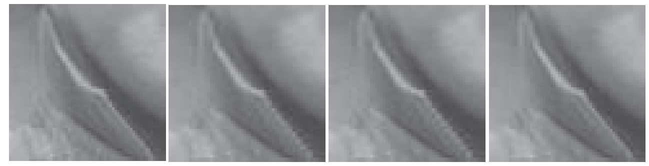
Having an initial estimation of all 8-neighboring pixels, NEDI-8 is implemented to get extra information from 4 directions in the second round. The model parameters can be directly estimated from its HR pixels in this round. Therefore, the overfitting problem of NEDI-4 is lessened while considering more edge orientations. For the sake of interpolation consistency, NEDI-8 is applied to the pixels of type 3, 2, and 1 as in this order. The fourth-order model parameters $\boldsymbol{h^4}$ and eighth-order model parameters $\boldsymbol{h^8}$ for HR scale are shown in Fig. 4b. The optimal $\boldsymbol{h^8}$ is similarly calculated by (8) where $\boldsymbol{y}$ is the vector of all HR pixels in W and A^8 is a $8 \times K$ matrix whose k^{th} column is composed of the 8-neighboring pixels of HR pixels. The final reconstructed frame is formed by

$$I'(m,n) = \begin{cases} P''(m,n) & \text{if } \alpha_m(m) = 0 \text{ and } \alpha_n(n) = 0 \\ \hat{P}'_h(m,n) & \text{otherwise} \end{cases}$$

4. SIMULATION RESULTS

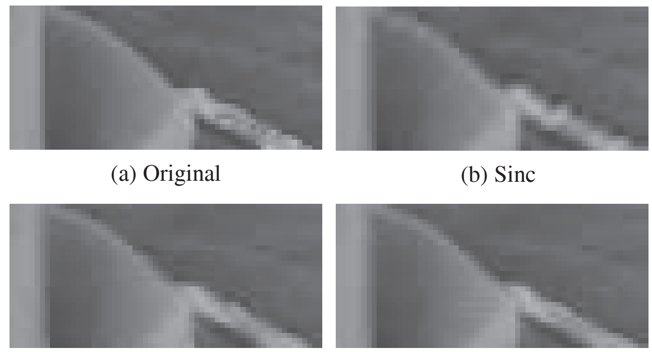
4.1. High order edge directed interpolation

Simulations are performed to compare the proposed high order edge-directed interpolation with other interpolation methods for a wide range of data in different formats. For the NEDI-6 case, original frames are downsampled by 2 in the horizontal direction. The downsized frames are then interpolated using bicubic, sinc, and the proposed NEDI-6 interpolation. Note that other interpolation methods, such as [9], can only be applied in cases of downsampling by ratio of 2 in both directions. A particular result is shown in Fig. 6 for a zoomed part of the 1^{st} frame of the Foreman sequence. The PSNR values of the interpolated frames using bicubic, sinc and NEDI-6 interpolation are $38.62\ dB$, $38.56\ dB$ and $39.22\ dB$, respectively. These results validate the effectiveness of NEDI-6 for edge-directed interpolation, since less jerkiness and higher PSNR is attained compared to the other methods.



(a) Original (b) Bicubic (c) Sinc (d) Nedi-6

Fig. 6. Comparison of NEDI-6 to other methods.



(a) Original (b) Sinc

(c) Nedi-4 (d) Nedi-8

Fig. 7. Comparison of NEDI-8 to other methods.

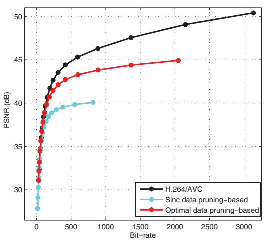
For the NEDI-8 case, the comparison is performed for bicubic, sinc, NEDI-4 and the proposed NEDI-8. To enhance the pixels near the frame borders in the proposed NEDI-8, the frame is expanded by reflecting these pixels over the borders. PSNR values are shown in Table 1 for sequences in different formats. To perform a fair comparison to other methods that use bilinear interpolation for pixels near the borders, pixels at 5 lines or fewer away from the border are not counted for the PSNR computation. The visual results for a selected part of Foreman sequence are shown in Fig. 7. As shown in Fig. 7b, the result using the sinc-based interpolation has a lot of jerkiness. While the NEDI-4 interpolation has significant less jerkiness, the interpolated frame in Fig. 7c still shows jerkiness along the strong edges. Because NEDI-4 only uses pixels of 2 directions, artifacts can be observed at the intersections of more than 2 edges. On the other hand, the NEDI-8 interpolated frame in Fig. 7d achieves the best quality with least jerkiness. Using pixels in 4 directions, the NEDI-8 interpolation also has less artifacts at the intersection of more than 2 edges. With respect to objective quality, the proposed NEDI-8 has the highest PSNR values for all the sequences across different formats.

4.2. Data pruning-based compression

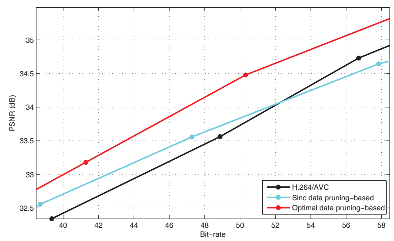
The data pruning approach is applied to video compression. An experiment is performed in which a GOP of 15 frames of Akiyo are pruned with the target $PSNR_{min} = 45\ dB$. As a consequence, the frame size is reduced from 720×480 to 480×320 lines. An H.264 codec is applied with the *IPPP* GOP structure and $QI = QP - 1 = [12, 40]$. The *LMSE* is averaged over the whole GOP, so that the

Table 1. PSNR comparison (in dB)

Sequences	Format	Bicubic	Sinc	Nedi-4	NEDI-8
Carphone	176×144	29.63	29.37	29.60	**30.28**
Foreman	352×288	33.36	33.25	33.10	**34.66**
Akiyo	352×288	34.70	34.59	35.15	**35.62**
Alphabet	720×480	29.71	29.62	30.39	**30.89**
Tiger	720×480	33.91	33.76	34.68	**35.13**
Average		32.262	32.118	32.584	**33.316**



(a) Whole R-D curves



(b) One zoomed in part

Fig. 8. Comparison results for R-D curves.

same lines are dropped for all the frames. In this way, the side information to determine the dropped lines is greatly reduced. The extra bit-rate is 1.2 $Kbps$ for the whole GOP, which is very small compared to the total bit-rate of the compressed bitstream. For comparison, the data pruning scheme is applied to the sequence down- and up-sized by 2×2 with the uniform sinc interpolation.

The rate-distortion (R-D) curves are shown in Fig. 8a (the whole curves) and in Fig. 8b (one zoomed in part). The results show that the R-D curve of the sinc data-pruned method is lower than the R-D curve of the optimal data pruning-based compressed sequence. Comparing to the R-D curve of the H.264/AVC compressed sequence, the one of the proposed method is better in the range of PSNR 32-37.5 dB. In this range, the PSNR improvement at the same bit-rate is around 0.3-0.7 dB.

Even having the same bit-rate and PSNR values, the reconstructed frames have less artifacts because they are compressed with smaller quantization step level QI and QP. Fig. 9 shows the comparison between the H.264/AVC compressed frame and the optimal data pruning-based compressed frame at the quantization level of 35 and 32, respectively. These sequences have nearly same bit-rate of 92 $Kbps$ and 94 $Kbps$ and same PSNR of 37.83 dB and 37.81 dB respectively for the H.264/AVC and the proposed data pruning-based compressed sequences. Results show that the proposed data pruning-based compressed frame in Fig. 9b has higher visual quality and less artifacts than the H.264/AVC compressed frame in Fig. 9a. This merit can be explained by the interpolation phase, which helps reducing the blocking and ringing artifacts, and the smaller quantization step level. The percentage of bit-rate saving of the optimal data pruning-based compressed sequence is 23-36% comparing to the H.264/AVC compressed sequence for the cases of using the same quantization step size.

Both PSNR curve and visual results validate the effectiveness of the proposed data pruning-based compression. The proposed algorithm requires a interpolation step in the data pruning and reconstruction phases, so the complexity of data pruning-based compression is higher than the normal compression. But the coding and decoding time of the proposed method decreases proportionally to the size reduction of the data pruned frame. All simulation results can be found at http://videoprocessing.ucsd.edu/∼dungvo/ICASSP09.html.

(a) H.264/AVC

(b) Optimal data pruning-based

Fig. 9. Comparison for H.264/AVC compression and optimal data pruning-based compression with same bit-rate and PSNR values.

5. CONCLUSIONS

The paper proposed a novel data pruning-based compression method to reduce the bit-rate. High order edge-directed interpolations are also discussed to include more surrounding pixels and to adapt to different data pruning schemes. The results show that these high order edge-directed interpolation methods help to reduce the jerkiness along strong edges and the artifacts at the intersection areas. The NEDI-6 for upsamling only rows can be also applied for deinterleaving. In a future work, instead of using only the pixels at odd indices, high order edge-directed interpolation methods may use more available pixels to estimate more accurately the model parameters. Additionally, the objective function in (2) may be extended to consider the coding efficiency of dropping these pixels to further improve the R-D curve.

6. REFERENCES

[1] “Advanced Video Coding for Generic Audiovisual Services,” 2005.

[2] N. Vasconcelos and F. Dufaux, “Pre and Post-filtering for Low Bit-rate Video Coding,” *IEEE Conf. on Image Process.*, vol. 1, pp. 291–294, Oct. 1997.

[3] A. Cavallaro, O. Steiger, and T. Ebrahimi, “Perceptual Prefiltering for Video Coding,” *IEEE Int. Symposium Intel. Multimedia, Video and Speech Processing*, pp. 510–513, Oct. 2004.

[4] K. Ratakonda and N. Ahuja, “POCS Based Adaptive Image Magnification,” *IEEE Conf. on Image Process.*, vol. 3, pp. 203–207, Oct. 1998.

[5] Y. Cha and S. Kim, “Edge-forming Methods for Color Image Zooming,” *IEEE Trans. Image Process.*, vol. 15, pp. 2315–2323, Aug. 2006.

[6] N. Mueller, Y. Lu, and M. N. Do, “Image Interpolation using Multiscale Geometric Representations,” *Proc. of SPIE Conf. on Electronic Imaging*, vol. 6498, Feb. 2007.

[7] N. Mueller and T. Q. Nguyen, “Image Interpolation using Classification and Stitching,” *IEEE Conf. on Image Process.*, Oct. 2008.

[8] M. Li and T. Q. Nguyen, “Markov Random Field Model-based Edge-directed Image Interpolation,” *IEEE Trans. Image Process.*, vol. 17, pp. 1121–1128, July 2008.

[9] X. Li and M. T. Orchard, “New Edge-directed Interpolation,” *IEEE Trans. Image Process.*, vol. 10, pp. 1521–1527, Oct. 2001.