DIRECTIONAL INTRA FRAME INTERPOLATION FOR HEVC COMPRESSED VIDEO

Xinwei Gao, Xiaopeng Fan, Min Gao, Debin Zhao
Harbin Institute of Technology, Harbin, China, 150001
Email: {xwgao.cs, fxp, mgao, dbzhao}@hit.edu.cn;

ABSTRACT

Image interpolation is one of the most elementary imaging research topics. A number of image interpolation methods have been developed and tested on uncompressed images in the literature. However, a lot of videos have already been stored or have to be transmitted in compressed format due to the storage limitation or the bandwidth limitation. The existed image interpolation methods may not be efficient when directly applied to compressed images or videos. Inspired by the success of the intra prediction in HEVC and the edge-directed image interpolation methods, a directional intra frame interpolation for HEVC compressed video is proposed. The main idea is to use the directional prediction information in compressed low-resolution video bitstreams to estimate the associated high-resolution video. For intra frames, the prediction direction information is taken into account as context in the directional interpolation. When a pixel is decompressed with a small prediction residual, the interpolation is performed along its block direction. The interpolation weight for each block direction is off-line trained by the Wiener filter based on the representative video sequences. For each pixel with a large prediction residual, a piecewise autoregressive model is used as a regularization term into the interpolation function. Extensive experiments demonstrate that the proposed method achieves better performance than the traditional methods such as Bicubic, KR, LAZA, NEDI and SAI.

Index Terms— interpolation, context, directional information, compressed video

1. INTRODUCTION

Image interpolation, which addresses the problem of rescaling an associated degraded low-resolution (LR) image to a clean and sharp high-resolution (HR) image, is one of the most elementary research topics in image processing. Image interpolation has various applications such as digital photography, video communication, satellite remote sensing, object recognition, medical analysis, and consumer electronics. One particular example is up converting video resolution from standard-definition (SD) to high-definition (HD). As HDTV sets has become widespread, there is a need to enhance SDTV signals which have been stored or have to be transmitted on the decoder side to match the quality and resolution of HDTV displays with low computational complexity. Considering the underlying image and video models in interpolation, the existing interpolation techniques can be categorized into three categories: data-invariant linear filters, local adaptive interpolation methods, and reconstruction based interpolation methods.

The data-invariant linear filters train the interpolation model using the whole image sample set. Representative linear filters are bilinear, cubic convolution [1] and cubic spline interpolators. These methods are global and with low computational complexity, which are widely used and favored for real-time applications. However, they may not be efficient for edges and texture regions and suffered from blurred edges and visual artifacts, such as ringing, zippering and jaggies.

As it is difficult to find a unique model with a good predictability for the whole image sample, the local adaptive interpolation methods [2-5] are proposed. They train the models by local image information and usually demonstrate better performance. A spatially adaptive interpolation algorithm called LAZA is proposed in [2], which performs interpolation along local edge directions. Li and Orchard propose a new edge-directed interpolation (NEDI) method [3], in which the linear regression model is used to estimate local covariance coefficients from a LR image and then to adapt the interpolation at the HR image based on the geometric duality between the LR and the HR covariance. Furthermore, Zhang and Wu propose the SAI algorithm [5], which learns and adapts varying scene structures using a locally linear regression model, and interpolates the missing pixels in a group by a soft-decision manner. Although these edge-guided local adaptive interpolation methods produce sharp edges and have less annoying artifacts than the data-invariant linear filters, the description ability of the edge-guided model is still limited. First, it is difficult to guarantee that the quality of interpolated image is best under the global point of view. Second, the size of the samples used to train a predictor may not be enough, which usually makes the predictor not sufficiently trained.

These two above methods only use single frame for interpolation; on the contrary, the reconstruction based interpolation methods [6-8, 13-16] for compressed video use multi-frames to generate a high quality frame. They formulate the process of image formation to build the relation between LR frames and HR frames with smoothness prior by using maximum a posteriori [6,7,13], projections onto convex sets (POCS)[8], or iterative back-projection (IBP)[14]. In [6], a stochastic framework is

proposed, in which quantization information as well as other statistical information about additive noise and image prior is utilized effectively. In [7], Bayesian resolution enhancement of compressed video is presented, which incorporates information from the bitstreams and fuses the super-resolution and post-processing problems. A motion-compensated, transform-domain super-resolution that directly incorporates the transform-domain quantization information is proposed [8]. Nonlocal means algorithm [15] and 3-D kernel regression [16] are also proposed to reconstruct HR frames by motion estimating without making use of the information in bitstreams. As was pointed in [6], all reconstruction based interpolation solutions are computationally expensive. These methods are also taken as image or video super-resolution reconstruction.

Usually the video is stored or transmitted in compressed format. The primary video compression standards have an important step: directional prediction. Intra frames are divided into intra blocks with different sizes to be encoded. For these intra blocks, directional prediction is applied. In this paper, a directional intra frame interpolation for HEVC [9] compressed video is proposed. The main idea is to use the directional prediction information in compressed low-resolution video bitstreams to estimate the associated high-resolution video. For intra frames, the prediction direction information is taken into account as context in the directional interpolation. When a pixel is decompressed with a small prediction residual, the interpolation is performed along its block direction. The interpolation weight for each block direction is off-line trained by the Wiener filter based on the representative video sequences. For each pixel with a large prediction residual, a piecewise autoregressive model is used as a regularization term into the interpolation function. Extensive experiments demonstrate that the proposed method achieves better performance than the traditional methods such as Bicubic, KR, LAZA, NEDI and SAI.

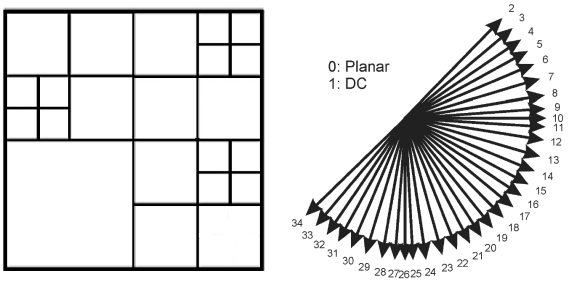

Fig.1 block segmentation and directional modes for the intra frame prediction.

2. DIRECTIONAL CONTEXT CLASSIFICATION

Directional structures are important to the human visual systems, which convey much of the information of image/video. Preserving directional structure is a challenge to the interpolation algorithms that reconstruct the high-resolution frames from the low-resolution counterpart. As indicated in [10], interpolation along edge direction is very effective. This is because, based on geometric constraint of edges, estimation along the edge orientation is optimal in the sense of best inferring unknown pixels. Inspired by the success of the intra prediction in HEVC and the edge-directed image interpolation methods, the directional context classification is presented.

In the block-based video coding standards, each intra block is coded using intra prediction without referring to any data outside the current frame. Intra prediction uses pixels from adjacent, previously coded block to predict the pixel values in the current block. We borrow the idea of intra frame direction prediction and take these 34 intra block directional modes as the directional contexts. Except for the DC and Planar modes, the other 32 modes correspond to different spatial directions as illustrated in Fig. 1. The directional prediction mode of a compressed LR block in bitstreams is used as a context to select the interpolation filter, which will be used for the associated HR block interpolation.

3. DIRECTIONAL FRAME INTERPOLATION

Let x be an LR decompressed intra frame. The vector $x \in Z^N$ consists of N LR pixel values in a given lexicographical order, where Z is an integer alphabet from which the pixel values are drawn. $\hat{y} \in Z^M$ is the original HR frame. $M = s^2 N$, $s > 1$. The formation of LR degraded frame x from $\hat{y}$ is modeled as:

$$x = HD\hat{y}, \tag{1}$$

where H is the operator of compression and quantization process and D is the downsampling operator. Reconstruction $\hat{y}$ from x is inherently an ill-posed inverse problem. The performance of the interpolation algorithm for compressed video depends on how well it can explore the prediction information as constraints. One efficient solution is to integrate the directional information, which is extracted from the bitstreams, into the estimation process.

The proposed method is illustrated by taking a single interpolation frame. Let y be the interpolated HR frame and x be its associated LR decompressed frame. As we all known, x can be divided into the prediction part and the residual part:

$$x = pre\_x + res\_x . \tag{2}$$

When a pixel $x(i)$ is decompressed with a small $|res\_x(i)|$, it means that the prediction direction can well describe the texture direction around the pixel $x(i)$. On the contrary, if the absolute value of $res\_x(i)$ is large, it means the edge of the pixel is not along the prediction direction accurately. In the following, we will detail the design of the intra pixel interpolation with different kinds of residuals respectively with 2×2 scaling as examples.

The pixel interpolation with small residual is defined as follows:

$$y(i) = \sum_{u \in S} W_C(u) x_i(u), \tag{3}$$

where $y(i)$ is the pixel to be interpolated in the HR frame. S is the spatial neighbourhood window in x. $W_C(u)$ is the weight of the interpolation filter selected by the context C. $C = mode(i)$ indicates the directional prediction mode of $y(i)$. It should be subject to:

$$\sum_{u\in S}\left|res\_x_i(u)\right| < T, \tag{4}$$

where T is the threshold used to judge the intensity of the residual.

In the following, we will detail the design of the pixel interpolation with large residual. In recent years, autoregressive model is very most popular and widely used for image and video interpolation, which can be locally computed from a LR frame by the least square model and well reconstructs the HR image. To model the structural similarity between the LR and HR frames, the pixel in the LR frame is assumed to share the same AR coefficients with the corresponding pixel in the HR frame:

$$W_{AR} = \underset{W}{\operatorname{argmin}} \sum_{i=1}^{n}\left(x(i) - \sum_{p\in W} W(u)_i x(u)\right)^2, \tag{5}$$

where W_{AR} is the AR coefficients vector, which is calculated on the LR image and is used to estimate the HR image. Different from the conventional AR model, we use a weighted AR model, which is shown:

$$W_{AR}^{\ k} = \underset{W}{\operatorname{argmin}} \sum_{i=1}^{n}\phi(i,k)\left(x(i) - \sum_{p\in W} W(u)_i x(u)\right)^2, \tag{6}$$

where $W_{AR}^{\ k}$ is the AR coefficients vector of the pixel $x(k)$, and $\phi(i,k)$ means the weight to describe the similarity texture between the patches with center of the pixel $x(k)$ and its neighbor $x(i)$.

$$\phi(i,k) = \frac{1}{Z(i)} e^{-\mu\left\|(P_i - P_i^{(k)})\bullet D(i,k)\right\|_2^2}, \tag{7}$$

$$Z(i) = \sum_{i\in S} e^{-\mu\left\|(P_i - P_i^{(k)})\bullet D(i,k)\right\|_2^2}, \tag{8}$$

In Eqs. (7) and (8), p_i and $p_i^{(k)}$ are the matrices whose elements are pixel values in the patches with center of $x(i)$ the k-th corresponding neighbor $x(k)$, $D(i,k)$ expresses the squared Euclidean distance, and μ is a constant to control the decay of the exponential function.

The pixel interpolation with large residual is defined:

$$y = \arg\min_{y}\left\{\begin{array}{l}\dfrac{\mu}{2}\left\|x - HDy\right\|_2^2 + \sum_{i=1}^{N}\left[x(i) - \sum_{u\in S} W_C(u) y_i(u)\right]^2 \\ +\sum_{i=1}^{N}\lambda(i)\left[y(i) - \sum_{p\in W} W_{AR}(p) y_i(p)\right]^2\end{array}\right\}. \tag{9}$$

The decompressed pixels in the HR interpolated frame are not changed, so $\left\|x - HDy\right\|_2^2$ should always be 0. The problem in Eq. (9) is converted to a least square problem

$$y = \arg\min_{y}\left\{\begin{array}{l}\sum_{i=1}^{N}\left[x(i) - \sum_{u\in S} W_C(u) y_i(u)\right]^2 \\ +\sum_{i=1}^{N}\lambda(i)\left[y(i) - \sum_{p\in W} W_{AR}(p) y_i(p)\right]^2\end{array}\right\}. \tag{10}$$

The closed form solution of (10) is

$$y = \arg\min_{y}\left\{\begin{bmatrix}A\\E\end{bmatrix} y - \begin{bmatrix}b\\0\end{bmatrix}\right\}^2 = \left(\begin{bmatrix}A\\E\end{bmatrix}^T\begin{bmatrix}A\\E\end{bmatrix}\right)^{-1}\begin{bmatrix}A\\E\end{bmatrix}^T\begin{bmatrix}b\\0\end{bmatrix}. \tag{11}$$

where A, E and b are made by x, W_C and W_{AR}.

4. TRAINING OF INTERPOLATION FILTER

In this section, we will show how to get the weights: W_C in Eq. (3) by least mean square error optimization.

First, a set of LR videos are generated by MPEG-B down-sampling [12]. Then the LR videos are intra-coded by HEVC. The weight matrix W_C is acquired by the training the pairs of the decompressed LR videos and the corresponding original HR videos. An estimated HR pixel $y(k)$ for context C can be obtained by

$$y(k) = \sum_{i=1}^{N} W_C(i) x_k(i), \tag{12}$$

where $x_k(i)$ is the corresponding pixels in x used for the interpolation $y(k)$. N is the number of $x_k(i)$. The main square error (MSE) between the estimated HR pixels and the corresponding original HR pixels $\hat{y}(k)$ in the train set for context C can be obtained by

$$e^2 = \sum_{k=1}^{N_C}\left[\hat{y}(k) - y(k)\right]^2 = \sum_{k=1}^{N_C}\left[\hat{y}(k) - \sum_{i=1}^{N} W_C(i) x_k(i)\right]^2, \tag{13}$$

where N_C represent the number of pixels of all training samples for context C. Then the optimal weights W_C can be computed by the MSE Wiener-Hopf equation [11]:

$$\frac{\partial e^2}{\partial W_C(i)} = \sum_{k=1}^{Nc} 2x_k(i)\left[\hat{y}(k) - \sum_{i=1}^{N} W_C(i) x_k(i)\right] = 0. \tag{14}$$

Finally, the weight W_C can be got by

$$\begin{bmatrix} W_C(1)\\ \vdots \\ W_C(N)\end{bmatrix} = \begin{bmatrix} \sum_{k=1}^{Nc} x_1(k)x_1(k) & \cdots & \sum_{k=1}^{Nc} x_1(k)x_N(k)\\ \vdots & \ddots & \vdots \\ \sum_{k=1}^{Nc} x_N(k)x_1(k) & \cdots & \sum_{k=1}^{Nc} x_N(k)x_N(k)\end{bmatrix}^{-1} \times \begin{bmatrix} \sum_{k=1}^{Nc} x_1(k)\hat{y}(k)\\ \vdots \\ \sum_{k=1}^{Nc} x_N(N)\hat{y}(k)\end{bmatrix}. \tag{15}$$

In the training on HEVC, we take these sequences: *RaceHorses* (416x240, 30Hz), *BlowingBubbles* (416x240, 50Hz), *BQMall* (832x480, 60Hz), *PartyScene* (832x480, 50Hz), *BasketballDrive* (1920x1080, 50Hz) and *Kimono* (1920x1080, 24Hz), 100 frames each sequence as the training set. All the frames are coded in intra frames with different QPs (QP=22, 27, 32, 37) by HEVC. So we finally got the interpolation filters for intra prediction modes.

5. EXPERIMENTAL RESULTS

In this section, we present experimental results of the proposed directional frame interpolation for compressed video, and compare our results to some representative techniques in the literature: Bilinear, Bicubic, the locally-adaptive zooming algorithm (LAZA), and the new edge-directed interpolation technique (NEDI). For thoroughness and fairness of our comparison study, we exploit some widely used sequences. First the MPEG-B down-sampling is used in our experiments (each frame is filtered and then down-sampled by the direct-subsampling method. The filter coefficient is set to be [2, 0, -4, -3, 5, 19, 26, 19, 5, -3, -4, 0, 2]/64) [12]. These LR video sequences are compressed by the encoder, and the proposed interpolation method is performed after decoding. In our experiments, the size of interpolation filter weight is a 16 steps (4×4) filter, which is adaptive to the intra prediction mode. The threshold T is set to be 20. Since the original HR images are known in the simulation, we can compare the interpolation results with the original sequences and measure the objective qualities by PSNR. The test platform used is Inter(R) Xeon(R) CPU E5606 2.13GHz with eight cores, 8.0GB RAM. The widely used sequences on HEVC are taken as the testing set. They are not included in the training set. These video sequences are compressed by HEVC with different QPs in the form of all Intra frames.

We show the experiments of the intra frame interpolation. Table I tabulates the objective quality comparison by PSNR compared the six different methods when applied to the nine test sequences, which are coded by HEVC in all Intra frames. From Tables I, it can be observed that the proposed intra frame interpolation performs better than other methods. The average PSNR gain is 0.52dB compared to Bicubic. Compared to KR, the average PSNR gain is more than 0.7dB. Our method also outperforms the edge detection based local methods: LAZA and NEDI. The gain is 0.55dB compared to LAZA. The average gain is more than 0.6dB with NEDI. Compared to SAI, the average PSNR gain is 0.88dB.

6. CONCLUSIONS

In this paper, we propose a directional intra frame interpolation for HEVC compressed video. The main idea is to use the directional prediction information in compressed low-resolution video bitstreams to estimate the associated high-resolution video. For intra frames, the prediction direction information is taken into account as context in the directional interpolation. When a pixel is decompressed with a small prediction residual, the interpolation is performed along its block direction. The interpolation weight for each block direction is off-line trained by the Wiener filter based on the representative video sequences. For each pixel with a large prediction residual, a weighted autoregressive model is used as a regularization term into the interpolation function. Extensive experiments show that the proposed method achieves better performance than these traditional methods such as Bicubic, KR, LAZA, NEDI and SAI.

Table 1. The objective quality comparison by PSNR.

sequences	QP	[1]	KR	LAZA	NEDI	SAI	Ours
Basketball Pass 416x240 50Hz	22	30.85	30.94	30.83	30.53	31.07	**31.21**
	27	30.50	30.60	30.49	30.24	30.78	**30.84**
	32	29.79	29.88	29.78	29.61	29.80	**30.05**
	37	28.58	28.65	28.58	28.48	28.29	**28.73**
BQSquare 416x240 60Hz	22	22.41	22.54	22.40	22.19	22.71	**22.79**
	27	22.34	22.47	22.33	22.13	22.64	**22.72**
	32	22.15	22.28	22.15	21.96	22.43	**22.49**
	37	21.66	21.77	21.66	21.51	21.86	**21.91**
RaceHorse 832x480 30Hz	22	29.45	29.61	29.42	29.41	**29.72**	29.61
	27	29.14	29.31	29.11	29.12	29.39	**29.43**
	32	28.42	28.58	28.41	28.43	28.55	**28.61**
	37	27.17	27.29	27.17	27.22	27.09	**27.31**
Vidyo1 1280x720 60Hz	22	36.03	36.76	35.99	35.92	34.68	**37.11**
	27	35.38	36.00	35.35	35.27	34.56	**36.24**
	32	34.18	34.64	34.17	34.03	33.15	**34.75**
	37	32.34	32.60	32.34	32.21	30.89	**32.63**
Vidyo4 1280x720 60Hz	22	35.99	36.33	35.93	35.45	33.99	**36.80**
	27	35.26	35.56	35.22	34.86	32.83	**35.92**
	32	34.06	34.28	34.04	33.84	31.76	**34.50**
	37	32.37	32.50	32.36	32.26	30.99	**32.60**
Vidyo3 1280x720 60Hz	22	34.15	34.66	34.10	34.14	34.18	**35.39**
	27	33.65	34.10	33.61	33.64	33.83	**34.68**
	32	32.63	32.95	32.60	32.52	32.26	**33.32**
	37	30.95	31.15	30.94	30.80	30.36	**31.29**
BQTerrace 1920x1080 60Hz	22	28.25	28.73	28.24	28.55	29.12	**29.47**
	27	28.03	28.49	28.02	28.32	28.80	**29.15**
	32	27.51	27.91	27.50	27.77	28.07	**28.43**
	37	26.49	26.78	26.49	26.68	26.83	**27.09**
Cactus 1920x1080 50Hz	22	32.06	32.36	32.00	32.14	32.31	**32.86**
	27	31.59	31.87	31.54	31.67	31.91	**32.25**
	32	30.63	30.86	30.61	30.71	30.63	**31.10**
	37	29.14	29.28	29.13	29.17	28.92	**29.38**
ParkScene 1920x1080 24Hz	22	33.86	33.86	33.74	33.47	33.97	**34.40**
	27	33.04	33.07	32.96	32.76	33.04	**33.42**
	32	31.56	31.61	31.52	31.42	31.41	**31.76**
	37	29.55	29.58	29.54	29.50	29.26	**29.62**
average		30.31	30.55	30.28	30.22	30.05	**30.83**

7. ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation of China (NSFC) under grants 61272386 and 61390513, the Program for New Century Excellent Talents in University (NCET) of China (NCET-11-0797), and the Fundamental Research Funds for the Central Universities (Grant No.HIT.BRETIII.201221).

8. REFERENCES

[1] R.G. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Transactions on Acoustics, Speech, Signal Processing*, vol. 29,no. 6, pp. 1153-1160, Dec. 1981.

[2] S. Battiato, G. Gallo, and F. Stanco, "A locally-adaptive zooming algorithm for digital images," *Image and Vision Computing*, vol. 20, no. 11, pp. 805–812, Sept. 2002.

[3] X. Li and M.T. Orchard, "New edge-directed interpolation," *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1521-1527, Oct. 2001.

[4] W. S. Tam, C. W. Kok and W. C. Siu, "Modified edge-directed interpolation for images" *Journal of Electronic Imaging*, vol. 19(1), pp. 013011, Jan–Mar. 2010.

[5] L. Zhang, X. Wu, "Image Interpolation by Adaptive 2-D Autoregressive Modeling and Soft-Decision Estimation," *IEEE Transactions on Image Processing*, vol.17, no. 6, pp. 887-896, June. 2008.

[6] B.K. Gunturk, Y. Altunbasak, and R.M. Mersereau, "Super-resolution reconstruction of compressed video using transform-domain statistics," *IEEE Transactions on Image Processing*, vol. 13, no. 1, pp. 33-43,Jan. 2004.

[7] C. A. Segall, A. K. Katsaggelos, R. Molina, and J. Mateos, "Bayesian resolution enhancement of compressed video," *IEEE Transactions on Image Processing*, vol. 13, no. 7, pp. 898-911, Jul. 2004.

[8] Y. Altunbasak, A. J. Patti, and R. M. Mersereau, "Super-resolution still and video reconstruction from MPEG-coded video," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 12, No. 4, pp. 217-226,Apr. 2002.

[9] G. J. Sullivan, J. R. Ohm, W. J. Han and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 22, No. 12, pp. 1649-1668, Dec. 2012.

[10] X. Liu, D. Zhao, R. Xiong, S. Ma, W. Gao, and H. Sun, "Image Interpolation via Regularized Local Linear Regression," *IEEE Transactions on Image Processing*, vol.20, no. 12, pp. 3455-3469, Dec. 2011.

[11] F. Jin, P. Fieguth, L. Winger, and E. Jernigan, "Adaptive Wiener filtering of noisy images and image sequences," *IEEE International Conferenceon Image Processing*, vol. 3, pp. III-349-52, Sept. 2003.

[12] "Spatial scalability filters," ISO/IEC TC1/SC29/WG11 and ITU-T SG16 Q.6, July. 2005.

[13] S.Belekos, N.Galatsanos, A. Katsaggelos, "Maximum a posteriori video super-resolution using a new multichannel image prior," *IEEE Transactions on Image Processing*. vol. 19, no. 6, pp. 1451–1464, Jun. 2010.

[14] B.C. Tom, A. Katsaggelos, "Resolution enhancement of monochrome and color video using motion compensation," *IEEE Transactions on Image Processing*. vol. 10, no. 2, pp. 278–287, Feb. 2001.

[15] M. Protter, M. Elad, H. Takeda, P. Milanfar, "Generalizing the non-local-means to super-resolution reconstruction," *IEEE Transactions on Image Processing*. vol. 18, no. 1, pp. 36–51, Jan. 2009.

[16] H. Takeda, P. Milanfar, M. Protter and M. Elad, "Super-resolution without explicit subpixel motion estimation," *IEEE Transactions on Image Processing*. vol. 19, no. 9, pp. 1958–1975, Sep. 2009.

[17] H. Takeda, S. Farsiu and P. Milanfar, "Kernel Regression for Image Processing and Reconstruction," *IEEE Transactions on Image Processing*. vol. 16, no. 2, pp. 349–366, Feb. 2007.