Non-causal Temporal Prior for Video Deblocking

Deqing Sun[1] and Ce Liu[2]

[1]Harvard University*, [2]Microsoft Research New England

Abstract. Real-world video sequences coded at low bit rates suffer from compression artifacts, which are visually disruptive and can cause problems to computer vision algorithms. Unlike the denoising problem where the high frequency components of the signal are present in the noisy observation, most high frequency details are lost during compression and artificial discontinuities arise across the coding block boundaries. In addition to sparse spatial priors that can reduce the blocking artifacts for a single frame, temporal information is needed to recover the lost spatial details. However, establishing accurate temporal correspondences from the compressed videos is challenging because of the loss of high frequency details and the increase of false blocking artifacts. In this paper, we propose a *non-causal temporal prior* model to reduce video compression artifacts by propagating information from adjacent frames and iterating between image reconstruction and motion estimation. Experimental results on real-world sequences demonstrate that the deblocked videos by the proposed system have marginal statistics of high frequency components closer to those of the original ones, and are better input for standard edge and corner detectors than the coded ones.

1 Introduction

Increasing amounts of video data are captured and shared everyday with the emergence of smart phones and mobile devices. Due to high data volume and bandwidth limit, storing and transmitting videos require low bit-rate compression, which introduces visually unpleasing coding artifacts to compressed videos. Fig. 1 shows some snapshots of typical real-world videos from Youtube and Skype. Compressed videos can suffer from severe image content distortion: high frequency image details are lost while artificial block boundaries appear. The compression artifacts are visually disruptive and can cause problems to vision algorithms that are primarily designed for uncompressed images. Therefore, it is desirable to remove or reduce these compression artifacts in compressed videos.

Post-processing coded videos at the decoder end is a promising solution to alleviate the conflict between bit rate reduction and image quality preservation, as this method does not change existing codec structure. Current coding standards divide an image into non-overlapping blocks and code each block individually, resulting in loss of correlation between neighboring blocks. Previous work [14, 18, 19] has relied on spatial smoothness priors to reduce blocking artifacts without blurring the major edges. Temporal information has been exploited to enhance the image details, but the improvement

*The work was performed while the first author was at Brown University.

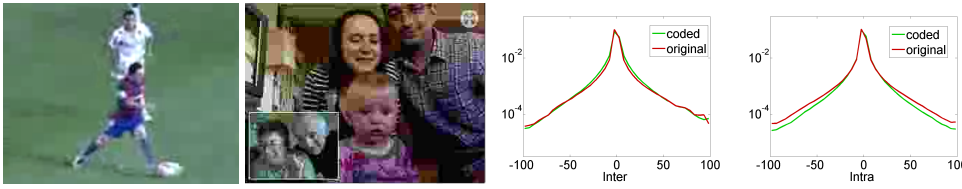


Fig. 1. Left two: snapshots of videos from Youtube and Skype. Compression artifacts, such as the artificial block boundaries and the absence of image details, severely degrade the video quality. It is desirable to recover the original videos from the compressed videos. **Right two**: histograms of neighboring pixel difference across (inter) and inside (intra) the coding blocks, obtained using 576 frames from 18 sequences. In the coded frames, artificial edges are introduced near the block boundaries, while details are lost inside blocks. Video deblocking should not only reduce the blocking artifacts but also enhance the lost details within the coding blocks.

is minor [12, 22]. Current video coding standards use block matching to reduce data redundancy, leaving little room for improvement by these block matching based methods.

Block matching is a coarse-level correspondence and therefore, does not fully remove temporal redundancy in the coding process. This phenomenon leaves room for further improvement through dense motion representation such as optical flow, which provides much finer correspondences than block matching, and further allows for exploiting remnant redundancy to help recover lost details.

Nevertheless, estimating optical flow on highly compressed videos is challenging. Typical motion estimation methods rely heavily on fine image details for accurate matching but many details are missing in the coded images. In addition, compression artifacts are false high frequency signals, which can disturb flow estimation algorithms. Inaccurate motion may deteriorate the quality of estimated videos (see Section 5).

In this paper, we study the video deblocking[1] problem in-depth and make the following contributions. First, we measure the statistical difference between coded videos (*i.e.* decompressed frames from a codec) and original uncompressed videos, and find that the coded ones have less details within the blocks. We try the image analogy approach to predict the lost high frequency components form low frequency ones but find this approach fails to recover the original videos. Second, we develop a non-causal temporal prior model that simultaneously estimates both the flow fields and the original video. Third, we study the optimum bit rate at which our system can get maximum performance gain. Fourth, the deblocked images have marginal statistics closer to the original images and are better input for standard edge and corner detectors than the coded images.

We also study the potential benefits of estimating optical flow at the encoder side. We find that, for scenes with simple motion, estimating the motion and spending a few extra bits to compress and transmit the motion can bring significant improvement for a postprocessing method. This suggests new doors for more efficient video compression.

[1] Although it has been commonly referred to as "deblocking" for reducing compression artifacts (including both JPEG and MPEG), our goal is not only reducing blocking artifacts but also recovering fine image details. We focus on the MPEG2 codec in our paper, and in theory our system can generalize to the more complicated MPEG4 codec.

2 Previous Work

The fundamental problem behind video deblocking is efficient image and video representation, which has been extensively studied in both coding and vision communities.

The fact that there are many repeated patterns, both spatially and temporally, makes it possible to compress digital images and videos [21]. The essence of compression is to predict a patch using other patches in a spatial or temporal neighborhood, and to discard the high-frequency components of the residual, as human eyes are less sensitive to high frequencies. In popular codecs such as MPEG2 and MPEG4 (including H.264), block matching has been widely used to predict patches in the current frame from neighboring frames. However, such coarse-level motion representation does not fully remove redundancies in videos, and leaves room for improvement by finer motion representations to exploit the remaining redundancy.

In computer vision, statistical image and video models have been studied to solve inverse inference problems. Sparse priors [24], non-local regularization [7], and high-order Markov Random Field (MRF) [23] can achieve high-quality image restoration results when the distortion process is known, such as denoising and deblurring. Contrary to block matching in the coding community, temporal correspondences are often described by more flexible representations such as optical flow fields [6, 10, 27]. Reliable optical flow estimation has enabled such successful applications as video denoising [16], super resolution [17], and high-speed video acquisition [20].

These prior models from the vision community can also be applied to video deblocking. However, we need to develop new models too. For example, instead of imposing spatial sparse prior on a static image [17]. We assume densely connected temporal prior for the whole video, which allows fast propagation of information over time. The formulation of [17] solves for flow between the reconstructed and the input images. Our formulation solves for flow between reconstructed images and avoids the difficult problem of using coding information in flow estimation. Our image reconstruction algorithm directly uses the coding information.

To remove JPEG blocking artifacts in a single image, researchers have used advanced image prior models developed in the vision community [14, 26] . These methods can reduce the blocking artifacts and preserve major image structures, but cannot recover the lost image details. Typically, image prior models may be able to preserve image details present in the input image, but not able to synthesize new details.

Repetitive measurements of objects in videos make it possible to recover the lost details. Although each individual frame may contain few details after compression, accumulating information over time can enhance details in every frame. To do so we need precise alignment of the images, which requires fine-level motion representation such as per-pixel optical flow.

Unfortunately, existing video deblocking methods do not use such fine-level motion representations. The deblocking filtering recommended by the MPEG standard adaptively performs low-pass filtering across the coding block boundaries [3, 30] on a frame-by-frame basis. Meier *et al.* [18] proposed to reduce the blocking artifacts in the video by repeatedly applying a single-frame deblocking method. Robertson and Stevenson [22] exploited temporal information for video deblocking but temporal correspondences are established by block matching. Li and Delp [12] proposed a 3D MRF for

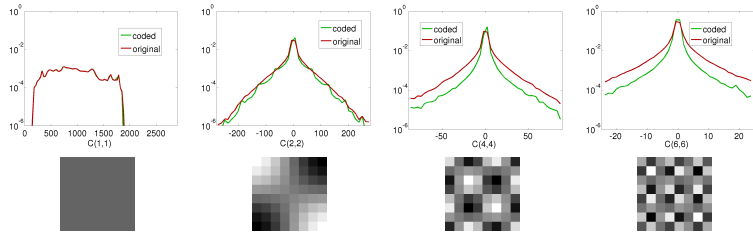


Fig. 2. Histograms (top) of the DCT coefficients $C(1,1)$ (DC), $C(2,2)$, $C(4,4)$, and $C(6,6)$, together with the corresponding DCT basis images (bottom), obtained using 576 frames from 18 sequences. The coded images have much fewer middle and high frequency DCT coefficients than the original images.

blocking artifact reduction with the temporal neighborhood also established by block matching. Jin *et al.* [11] used the magnitude of the image gradients, called "optical flow magnitude", to measure the surface smoothness and minimize this measure to reduce blocking artifacts. However, the method neither computes optical flow nor uses temporal information.

To measure the quality of deblocked images, Minami and Zakhor [19] proposed to use the smoothness across the coding block boundaries. However, this measure does not capture the loss of true image details. We study the statistical differences between the coded and the original images both between and inside the coding blocks and find that both measures are necessary criteria for designing a deblocking system.

3 Observations about MPEG Video Coding

3.1 Brief Introduction

Transform coding has been demonstrated as being effective for compressing both still and moving pictures. Due to its energy concentration ability, block discrete cosine transform (DCT) has been widely adopted in several coding standards, such as JPEG [29] and MPEG2 [25]. The encoder divides an image into non-overlapping blocks and forms a prediction of each block using neighboring images or other blocks in the same image. Let vector I be vectorized representation of a block in the original image and I^{pred} its prediction. To remove temporal redundancy, the encoder subtracts the prediction from the original image block and transfroms the the residual block into the DCT domain

$$C = \mathbf{T}(I - I^{pred}), \tag{1}$$

where $\mathbf{T}$ is the forward DCT transformation matrix. To save bits, the encoder performs quantization to each DCT coefficient $C(u,v), 0 \leq u,v \leq 7$, and the decoder performs dequantization. The whole process can be described as

$$C_q(u,v) = \left[\frac{C(u,v)}{Q(u,v)}\right] Q(u,v), \tag{2}$$

where $[*]$ means rounding operation, and $Q(u,v)$ is the quantization step size for the (u,v)th coefficient. We can extract the quantized coefficients and the quantization step

sizes for every block from the coding bit stream. These two together define an interval, or a quantization constraint set, which the original coefficients before quantization must belong to. Performing the inverse transform on the quantized coefficients and adding the result to the prediction, we obtain the decompressed (*a.k.a.* coded) block as

$$J = \mathbf{T}^{-1} C_q + I^{pred}, \tag{3}$$

where $\mathbf{T}^{-1}$ is the inverse DCT transformation matrix.

For most image blocks, high frequency coefficients have small magnitude and are usually truncated by the quantization operation. The reconstructed blocks hence have less high frequency components than the original ones. Because each block is coded independently, false edge may arise across the blocks boundaries.

3.2 Statistics of Original and Coded Sequences

We collected 18 standard videos from the coding community[1] and compressed them using the official MPEG2 software [2] at 0.5 Mbps with the main profile setting. We then computed the statistics of the original and coded sequences. First, coding introduces artificial high frequency components across the block boundaries, while reducing the high frequency signals within each coding block, as shown in Fig. 1. Although the increase of high frequency blocking artifacts is more obvious and receives more attention, the recovery of the lost high frequency components within each coding blocks has been more or less neglected.

To further study the loss of high frequency components, we computed the histograms of the DCT coefficients of the original and the coded sequences. Compared to the original, the coded sequences have similar histograms for the low-frequency coefficients, but the magnitude of middle and high frequency coefficients are significantly reduced, as shown in Fig. 2. Fig. 4 plots the ratio of the coded DCT coefficient magnitude over that of the original sequences. For most middle and high frequency DCT coefficients, the ratio is below 1. All these statistics suggest that the coding system significantly reduces the middle and high frequency signal within blocks, while introducing unwanted high frequencies across blocks.

3.3 A Naïve Approach

An simple and intuitive idea is to restore the coded blocks using a data-driven, learning based approach in the spirit of image analogies [9]. For a given bit rate, we can obtain pairs of coded and original sequences. For a new coded sequence, we can match the coded blocks in the database and transfer the corresponding original blocks to restore the images. Specifically, for every selected video, we collect 8×8 image blocks from the rest 17 video compressed at the same bit rate, perform PCA on these blocks, and build a K-D tree [5, 13] using the top 20 PCA coefficients for fast searching neighboring coded blocks. For each block in the input video, we find its approximate k nearest neighboring blocks via the pre-built K-D tree, and add the weighted average of the difference between the original image blocks and the coded blocks to the block to restore. As shown in Fig. 5, this approach does introduces high frequency to the image, but the improvement is small, as shown in Section 5.

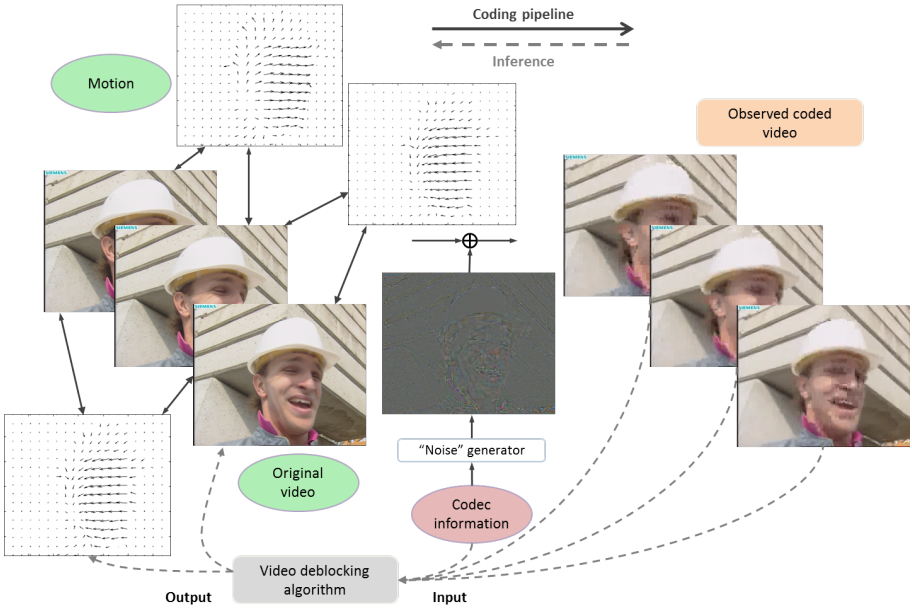


Fig. 3. The diagram of video deblocking. The original frames are densely connected to both the nearest temporal neighbors and those far away. The coded images are generated by adding compression artifacts to the original frames. Our video deblocking system takes compressed video and outputs both the decompressed video and the inter-frame flow fields.

The results are actually not surprising by afterthought. The DCT transform is well known for its high decorrelation ability. The correlation between the low frequency and high frequency coefficients tends to be low. Hence it it is hard to find the truncated high frequency coefficients back using the little correlated low frequency coefficients. We should exploit information from multiple images which provide repeated observations of the same patterns.

4 A Non-causal Temporal Prior Model

Given a compressed video sequence $\{J_i\}$, our goal is to recover the original sequence $\{I_i\}$ using temporal information from neighboring frames. We design a non-causal system to jointly estimate the original sequence $\{I_i\}$ and the correspondences $\{w_{ij}\}$ between neighboring frames. Fig. 3 illustrates the model of how the compressed video is generated and the corresponding graphical model is given in Fig. 4. Note that both directed and undirected edges exist in the graph.

For computational reasons, we use Bayesian MAP to find the optimal solution

$$\{\{I_i\}^*, \{w_{ij}\}^*\} = \underset{\{I_i\},\{w_{ij}\}}{\operatorname{argmax}} p(\{I_i\}, \{w_{ij}\}|\{J_i\}), \tag{4}$$

where the posterior is the product of prior and likelihood:

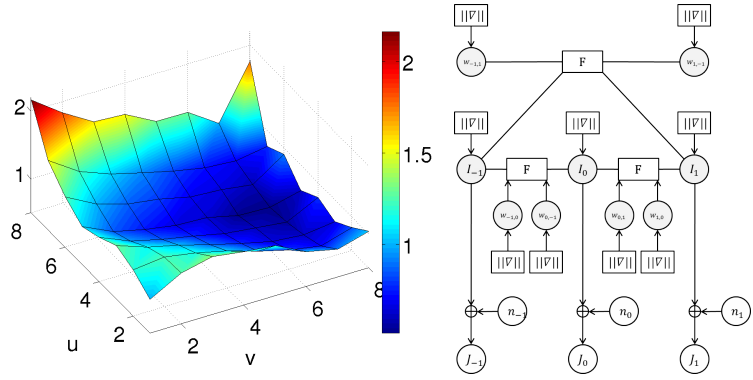


Fig. 4. Left: ratios of DCT coefficient magnitude between the coded and original images. Frequency increases from top left to bottom right. The ratio for the DC coefficient $(1,1)$ is 1. The coded coefficients roughly preserve the energy at low frequency bands but have loss in the middle and high frequency ones. The increase in the highest vertical frequency (last row) results from the "field"-based motion prediction by MPEG2 (please refer to supplementary materials for further details). **Right**: The graphical model of video deblocking. We allow non-adjacent neighbors to be temporal neighbors. The circular nodes are variables (vectors), whereas the rectangular nodes **F** is the warping matrices according to the flow field. The end of the directional edges indicate the constraint after transformation.

$$p(\{I_i\},\{w_{ij}\}|\{J_i\}) \propto p(\{w_{ij}\}) \cdot p(\{I_i\}|\{w_{ij}\}) \cdot \prod_i p(J_i|I_i). \tag{5}$$

We use a sparsity prior to model the correspondences between images:

$$p(\{w_{ij}\}) \propto \prod_i \prod_{j\in\mathcal{N}_i} \exp\left\{-\lambda\left(\left\|\nabla u_{ij}\right\|+\left\|\nabla v_{ij}\right\|\right)\right\}, \tag{6}$$

where the set $\mathcal{N}_i$ contains the temporal neighbors of the ith frame, u_{ij} and v_{ij} are the horizontal and vertical components of the flow field w_{ij}, respectively, and $\left\|*\right\|$ denotes the L1 norm. We allow non-adjacent frames to be neighbors and define the temporal neighborhood size to be the largest time interval between i and its neighbors. Given the correspondences, the distribution for the original video sequence is

$$p(\{I_i\}|\{w_{ij}\}) \propto \prod_i \exp\{-\eta_i\left\|\nabla I_i\right\|\} \cdot \prod_{j\in\mathcal{N}_i} \exp\left\{-\eta_{ij}\left(\left\|I_i-\mathbf{F}_{w_{ij}}I_j\right\|+\left\|I_j-\mathbf{F}_{w_{ji}}I_i\right\|\right)\right\}, \tag{7}$$

where w_{ij} is the correspondence from the ith frame to the jth frame and $\mathbf{F}_{w_{ij}}$ is the warping operator. Note that I_i is related to I_j by both w_{ij} and w_{ji}.

We model the coding distortion by the Laplacian distribution

$$p(J_i|I_i) \propto \exp\{-\theta_i||I_i - J_i||\}. \tag{8}$$

4.1 Image Reconstruction

Given the current estimates of the correspondence $\{w_{ij}\}$ and all the other frames $\{I_j\}_{j\neq i}$, we estimate the original image I_i by solving

$$I_i^* = \underset{I_i}{\operatorname{argmin}}\, \theta_i \left\| I_i - J_i) \right\| + \eta_i \left\| \nabla I_i \right\| + \sum_{j \in \mathcal{N}_i} \eta_{ij} \left(\left\| I_i - \mathbf{F}_{w_{ij}} I_j \right\| + \left\| I_j - \mathbf{F}_{w_{ji}} I_i \right\| \right), \quad (9)$$

within the quantization intervals for each 8×8 coding block

$$C_q(u,v) - \frac{Q(u,v)}{2} \leq C(u,v) < C_q(u,v) + \frac{Q(u,v)}{2}, 0 \leq u, v \leq 7. \quad (10)$$

To use gradient-based methods, we replace the L1 norm with a differentiable approximation $\phi(x^2) = \sqrt{x^2 + \epsilon^2}$ ($\epsilon = 0.001$), and denote $\Phi(|I|^2) = \{\phi(I^2(q))\}$. We optimize Eq. 9 by the iterated reweighted least square (IRLS) method [15] solving the following linear system:

$$\left[\theta_i \mathbf{W}_i + \eta_i \left(\mathbf{D}_x^T \mathbf{W}_s \mathbf{D}_x + \mathbf{D}_y^T \mathbf{W}_s \mathbf{D}_y \right) + \sum_{j \in \mathcal{N}_i} \eta_{ij} (\mathbf{I} + \mathbf{F}_{w_{ji}}^T \mathbf{W}_{ji} \mathbf{F}_{w_{ji}}) \right] I_i =$$

$$\theta_i \mathbf{W}_i J_i + \sum_{j \in \mathcal{N}_i} \eta_{ij} (\mathbf{I} + \mathbf{F}_{w_{ji}}^T \mathbf{W}_{ji}) I_j, \quad (11)$$

where the matrices $\mathbf{D}_x$ and $\mathbf{D}_y$ correspond to the x- and y- derivative filters and $\mathbf{I}$ is the identity matrix. IRLS iterates between solving the above least square problem and estimating the diagonal weight matrices $\mathbf{W}_i = \text{diag}(\Phi'(|I_i - J_i|^2))$, $\mathbf{W}_s = \text{diag}(\Phi'(|\nabla I_i|^2))$, and $\mathbf{W}_{ji} = \text{diag}(\Phi'(|\mathbf{F}_{w_{ji}} I_i - I_j|^2))$ based on the current image estimate. During the optimization process, the DCT coefficients of the restored image might fall out of the quantization intervals specified by the coded coefficients and the quantization step size, we project these coefficients to the closest points within the quantization intervals.

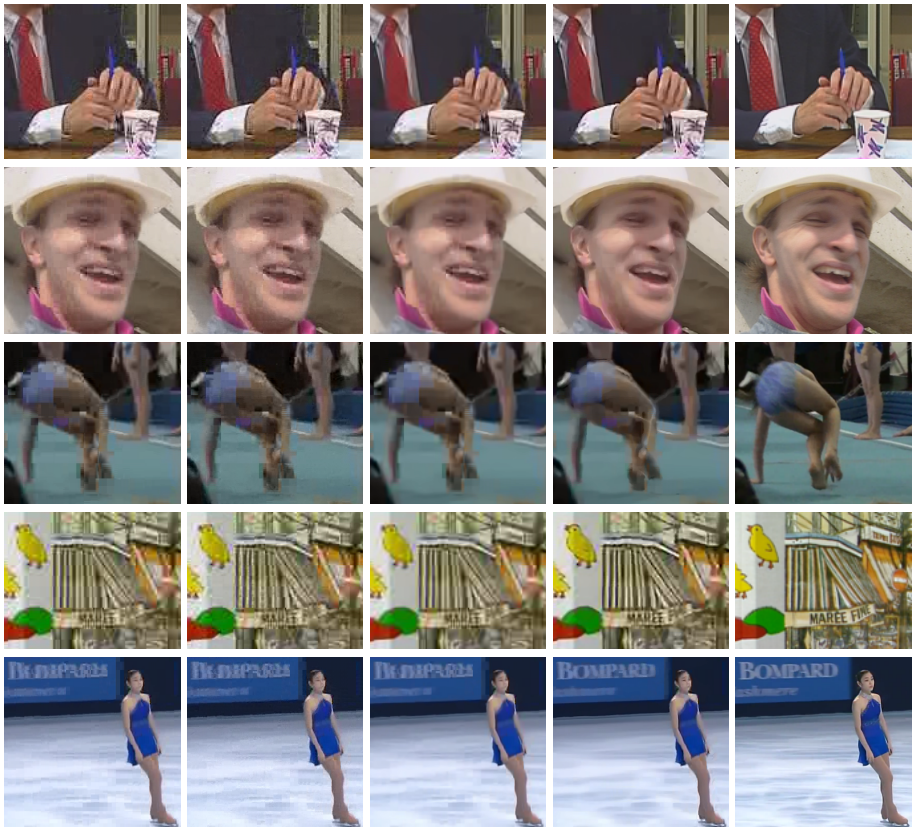
4.2 Motion Estimation

The deblocked images get closer to the original images with more iterations. Hence given the deblocked images, we estimate the flow field w_{ij} to establish the correspondence between I_i and I_j (the deblocked versions of J_i and J_j). We adopt a warping-based, incremental estimation process and solve the flow increment by the SOR method [6].

5 Experimental Results

We test the proposed method using several real-world video sequences compressed by the MPEG2 video codec. For multi-frame version, we set the temporal neighborhood size to be 3. One iteration of our method includes solving all the correspondences between neighboring frames and all the frames. Because there is no public software for video deblocking, we compare our system to two state-of-the-art video denoising methods, HQVD [16] and VBM3D [8], and the K-D tree based regression method that uses the top 7 nearest neighbors for prediction. We manually tune the parameters using the "yuna" and "city" sequences.

Table 1 summarizes the PSNR results. On average, the proposed method (*M-iter3) has 1.16dB gain over the coded sequences and 0.76dB over VBM3D. As shown in Fig. 5 and 6, the proposed method effectively reduces the blocking artifacts, while recovering good amount of image details. For example, the eyes and the mouth are more clear on "foreman"; the banner is almost readable on "calendar"; and the background letters are easily recognizable on "yuna".



(a) Coded (b) K-D tree (c) VBM3D [8] (d) Ours (e) Original

Fig. 5. Experimental results and comparison. From top to bottom are *pairs*, *foreman*, *gym*, *calendar* and *yuna*. **Please see the supplementary materials for corresponding videos, and view this figure on a computer screen to see the difference**. Our video deblocking system can reduce the coding artifacts and recover details missing in the coded frames.

Our C++ implementation takes about 3 hours to process the 32 frame 640×384 "yuna" sequence. Half of the time is on flow estimation and the other half on image reconstruction. Future work will address reducing the computation by better linear equation solvers [28] and GPU implementations.

Linear programming solver. Linear programming (LP) is another option to solve the image reconstruction problem within quantization constraints. We tested both the conjugate gradient (CG) solver with projection and the linear programming solver on single-image deblocking problem. The LP solver achieves a solution with nearly the same energy as the CG solver. However a 176×252 image took the MATLAB built-in LP solver about one hour but the CG solver less than 30 seconds in MATLAB. Therefore we choose CG with projection over LP in our system.

Comparison of motion. We are interested in how motion accuracy would affect deblocking results. Since the MPEG2 encoder estimates motion by block matching, we

(a) pairs (b) foremanA (c) foremanB (d) gym (e) calendar (f) yuna

Fig. 6. Closeup of Figure 5. From top to bottom: *coded*, *VBM3D [8]*, *our system*, and *original*.

Table 1. PSNR score of the luminance (Y) component, averaged over 32 frames. Proposed-S means the proposed method using a single coded frame; *M-iter$n$ means the proposed method using multi-frame model at the nth iteration. *M-MPEG2 is the proposed multi-frame system using the motion vectors from MPEG2, and *M-GT is using the "ground truth" motion estimated using the original images. VBM3D [8] uses all the 32 frames.

	Coded	K-D Tree	HQVD [16]	VBM3D [8]	Proposed-S	*M-iter0	*M-iter3	*M-MPEG2	*M-GT
yuna	32.90	32.35	33.24	33.36	33.00	34.18	**34.65**	33.21	34.90
city	31.24	30.43	29.93	31.44	31.28	32.05	**32.15**	29.87	32.53
gym	31.09	30.53	N/A	31.53	31.21	31.92	**32.21**	31.28	33.07
pairs	28.91	28.21	N/A	29.28	29.03	29.44	**29.65**	29.60	29.88
foreman	33.93	32.69	N/A	34.44	34.13	34.94	**35.20**	33.93	35.73

can establish correspondences using the block matching results. Using such correspondences (*M-MPEG2 in Table 1) produces results slightly better than the single-frame approach on most sequences, consistent with previous work [12, 22]. We can also compute the "ground truth" motion using the original video sequences. Using the "ground truth" motion in the multi-frame system (*M-GT in Table 1), we have about 0.4 dB improvement in PSNR over the motion estimated at the decoder. This improvement suggests that more accurate motion leads to better reconstruction.

Convergence. We study the effect of iterations and compare the quality of the estimated motion at each iteration w.r.t. the "ground truth". We find that the motion gets better and the quality of the reconstructed images increases with more iterations, as shown in Fig. 8. In addition, the curves suggest that our algorithm converges in about 3-5 iterations between motion and images. Note that we use coordinate descent to optimize a convex formulation and the algorithm should converge theoretically.

Optimum bit rate exists for video deblocking. With an infinitely high bit rate, there is no coding loss and video deblocking cannot help. With a zero bit rate, there is no information for video deblocking. Therefore, we expect video deblocking to produce maximum improvement for certain bit rates. To test this, we process the "foreman" sequence compressed at different bit rates. As shown in Fig. 10, the optimal bit rate for the proposed system is between 0.5 and 2 Mbps.

(a) Frame 1 (b) Frame 2 (c) Block matching (d) Optical flow (e) Color key

Fig. 7. Correspondence on one pair of the "yuna" sequence established by the MPEG2's block matching method (white blocks are usually "intra" blocks that do not have motion vectors and the alternating vertical lines in some blocks result from the field DCT coding by MPEG2) and by our optical flow-based system, color encoded as in [4]. Optical flow estimation provides more piecewise smooth correspondences than MPEG2's block matching method.

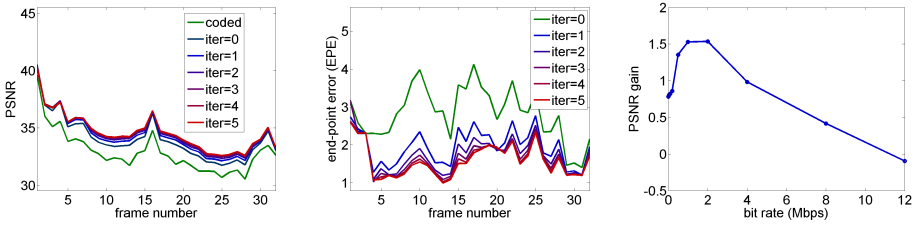


Fig. 8. Left: PSNR (higher, better) curves for "yuna" sequence during each iteration. Middle: with more iterations, the estimated correspondence become closer to that estimated using the original images, as indicated by the end-point error (lower better). The quality of reconstructed images is highly correlated with the quality of the estimated motion. Right: PSNR gain versus coding bit rate for "foreman". The proposed system tends to perform well when the bit rate is between 0.5 and 2 Mbps. When the bit rate is high, the images are of high quality and postprocessing helps little. When the bit rate is too low, the image quality is too low to make motion estimation work.

Recovery of image details. To further test how well the proposed method recovers the fine details, we compute the histograms of the DCT coefficients using the reconstructed images by the proposed method. As shown in Fig. 9, the histograms of the high frequency DCT coefficients of the reconstructed images are closer to the histograms of the original sequences than the coded sequence, both visually and numerically as measured by the KL distance. The results suggest that estimating and using optical flow can recover certain lost details.

Embedding learning-based methods to the proposed system. We are interested in whether more useful high frequency components can be introduced by using the learning-based approach. Hence we modify the data term of the proposed system and use the K-D tree regression output as the observation. We test this system using the "yuna" sequence but find little improvement in PSNR. After 3 iterations, the average PSNR is 34.54dB, almost the same as 34.65dB by the proposed system.

Message to the video coding community. We have seen the benefits of estimating optical flow at the decoder end. What benefits will we have if we estimate optical flow at the encoder and transmit it to the decoder? We generate a synthetic video sequence by circularly translating one "yuna" image to obtain a sequence with known translation motion. We compress this sequence by the MPEG2 encoder and Fig. 10 shows two

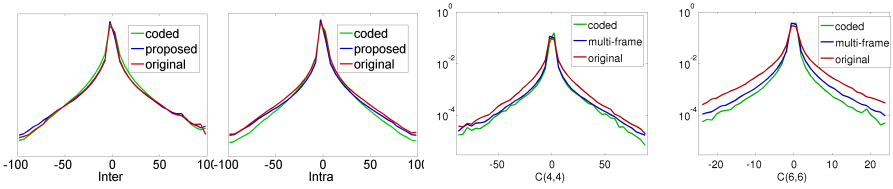


Fig. 9. Left to right: histograms of neighboring pixel difference across (inter) and inside (intra) the coding blocks and the DCT coefficient $C(4,4)$ and $C(6,6)$, obtained using 576 frames from 18 sequences. The histograms of the deblocked images by the proposed system are closer to those of the original images than those of the coded images. For the inter case, the K-L distance between the histograms of the original and the coded/proposed is 0.005/0.003; for the intra case, the numbers are 0.002/0.006. The increase in the intra case results from the smoothing of small gradient magnitude (sharp peak around zero), though the proposed method enhances the large gradient-magnitude signals. For $C(4,4)$, the distances are 0.043/0.014; for $C(6,6)$, the distances are 0.133/0.066.

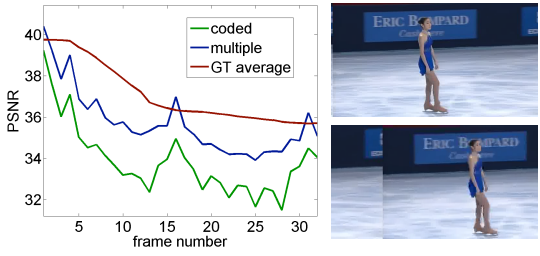
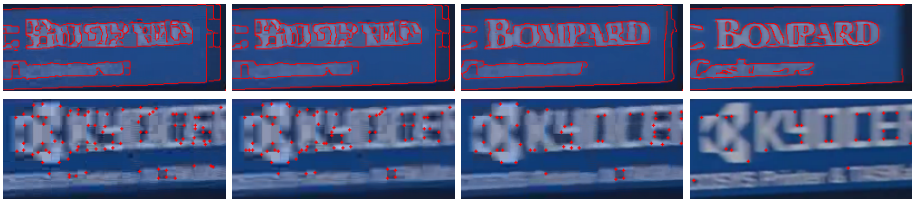


Fig. 10. Left: PSNR curves for compressing a translating sequence (5 pixels/frame). The proposed system produces results of higher quality than the coded images, but not as good as simple averaging processing with ground truth motion ("GT average"). **Right**: coded 1st and 32nd frames; note the quality degradation of the 32nd frame (better view on screen).

frames of the coded sequence. At the decoder, we iteratively average temporally neighboring frames along the translation motion and the simple averaging scheme produces better results than the proposed system for most frames. For this simple sequence, it costs only several bytes per frame to transmit the motion. This suggests that for sequences with simple motion (even in some part of the sequence), estimating and coding the motion may result in significant improvement at the decoder side.

Message to the vision community. What benefits will a good deblocking method bring to the vision community? Vision algorithms are usually developed without considering the coding process, but real-world videos are all compressed to some degree. Therefore, we study how different input images influence the low-level vision tasks.

We apply the MATLAB built-in Canny edge and Harris corner detectors on the coded, deblocked, and original images. We set the results obtained using the original images as the ground truth. Table 2 summarizes the F-score results. Using the deblocked images produces edge detection results closer to the ground truth. As shown in Fig. 11, the coded images cause false edges detected across the coding block boundaries, while the edge detection results with the deblocked images are closer to the ground truth.



(a) Coded (c) VBM3D [8] (d) Ours (e) Original

Fig. 11. Results of Canny edge (top) and Harris corner (bottom) detectors on "yuna". Most edges around the letters are correctly detected and much fewer false corners are detected using the deblocked images by our system.

Table 2. F score (larger the better) for edge and corner detection, averaged over 32 frames.

	edge detection					corner detection				
	yuna	*city*	*gym*	*pairs*	*foreman*	*yuna*	*city*	*gym*	*pairs*	*foreman*
Coded	0.380	0.695	0.517	0.811	0.735	0.242	0.516	0.367	0.422	0.568
VBM3D [8]	0.384	0.695	0.523	0.817	0.735	0.288	0.535	0.395	0.431	0.578
Proposed-M-iter3	**0.438**	**0.736**	**0.553**	**0.820**	**0.757**	**0.364**	**0.576**	**0.440**	**0.451**	**0.587**

Using deblocked images produces fewer false corners, too. Both Canny and Harris detectors have been widely used in computer vision and the improvement obtained by our system can benefit these tasks.

6 Conclusions and Future Work

We have studied the differences in marginal statistics between the coded and the original videos. The coded videos have extra unwanted high frequency content across block boundaries but lost details within the blocks. We have proposed a non-causal video deblocking system that simultaneously solves for the optical flow and the original video sequence. We find that reliable optical flow estimates provide important temporal information to recover fine image details, which in turn help refine the optical flow estimates. Our results suggest that a good video deblocking system can be a useful pre-processor for vision methods designed for uncompressed images. Our work also suggests the potential benefits of estimating optical flow and encoding the motion information for scenes with simple motion.

References

1. `http://media.xiph.org/video/derf`.
2. `http://www.mpeg.org/MPEG/video`.
3. *MPEG4 Verification Model, VM 18.0, 2001*. pp. 271-275.
4. S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *IJCV*, 92(1):1–31, 2011.
5. J. L. Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18:509–517, 1975.

6. T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, 2004.
7. A. Buades, B. Coll, and J. Morel. A non-local algorithm for image denoising. In *CVPR*, 2005.
8. K. Dabov, A. Foi, and K. Egiazarian. Video denoising by sparse 3d transform-domain collaborative filtering. In *EUSIPCO*, 2007.
9. A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In *SIGGRAPH*, pages 327–340, 2001.
10. B. Horn and B. Schunck. Determining optical flow. *AI*, 16:185–203, Aug. 1981.
11. X. Jin, S. Goto, and K.-N. Ngan. Optical flow based DC surface compensation for artifacts reduction. In *Picture Coding Symposium*, 2009.
12. Z. Li and E. Delp. MAP-based post processing of video sequences using 3-d huber-markov random field model. In *ICME*, 2002.
13. L. Liang, C. Liu, Y.-Q. Xu, B. Guo, and H.-Y. Shum. Real-time texture synthesis by patch-based sampling. *ACM Trans. Graph.*, 20(3):127–150, 2001.
14. A. Liew and H. Yan. Blocking artifacts suppression in block-coded images using overcomplete wavelet representation. *IEEE TCSVT*, 14(4):450–461, April 2004.
15. C. Liu. *Beyond pixels: exploring new representations and applications for motion analysis.* PhD thesis, MIT, 2009.
16. C. Liu and W. T. Freeman. A high-quality video denoising algorithm based on reliable motion estimation. In *ECCV*, 2010.
17. C. Liu and D. Sun. A Bayesian approach to adaptive video super resolution. In *CVPR*, 2011.
18. T. Meier, K. Ngan, and G. Crebbin. Reduction of blocking artifacts in image and video coding. *IEEE TCSVT*, 9(3):490–500, April 1999.
19. S. Minami and A. Zakhor. An optimization approach for removing blocking effects in transform coding. *IEEE TCSVT*, 5(2):74–82, April 1995.
20. D. Reddy, A. Veeraraghavan, and R. Chellappa. P2C2: Programmable pixel compressive camera for high speed imaging. In *CVPR*, 2011.
21. I. E. Richardson. *The H.264 Advanced Video Compression Standard*. Addison-Wesley, 2010.
22. R. L. Robertson, M. A.and Stevenson. Restoration of compressed video using temporal information. In *Proc. of SPIE*, 2001.
23. S. Roth and M. Black. Fields of experts: a framework for learning image priors. In *CVPR*, 2005.
24. L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4):259–268, 1992.
25. T. Sikora. MPEG digital video-coding standards. *Signal Processing Magazine, IEEE*, 14(5):82–100, Sep. 1997.
26. D. Sun and W.-K. Cham. Postprocessing of low bit-rate block DCT coded images based on a fields of experts prior. *IEEE TIP*, 16(11):2743–2751, 2007.
27. D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *CVPR*, 2010.
28. R. Szeliski. Locally adapted hierarchical basis preconditioning. *SIGGRAPH*, 2006.
29. G. K. Wallace. The JPEG still picture compression standard. *Commun. ACM*, 34:30–44, 1991.
30. T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the H.264/AVC video coding standard. *IEEE TCSVT*, 13(7):560 –576, July 2003.