An Efficient Video Super-Resolution Approach based on Sparse Representation

Farzad Toutounchi, Valia Guerra Ones, Ebroul Izquierdo
Multimedia and Vision Research Group, Queen Mary University of London
{f.toutounchi,v.guerra,e.izquierdo}@qmul.ac.uk

***Abstract*—Block-wise super-resolution methods, and in particular sparse representation based approaches, often focus on spatial upsampling of still images. Applying such models to videos is an extremely time consuming process due to the expensive sparse coding process for every block of each frame, and the conventional exhaustive overlapping blocks processing for reducing the blocking artifacts. In this paper, we introduce an approach that enables us to skip the sparse representation process for the static portions of a video by predicting the high resolution blocks from previously super-resolved frames, exploiting the significant amount of correlation between the adjacent frames in a video. Our approach is also enhanced by adaptive in-loop filters for removing the blocking and pixel-wise artifacts, replacing the overlapping blocks structure of super-resolution. Our method provides comparable results in terms of image quality with respect to the state of the art, and can reduce the processing time of the super-resolution methods by 63% in average. It enables the block based super-resolution methods to be applied in upsampling of videos to very high resolutions with a reasonable processing time.**

I. Introduction

Super-resolution (SR) techniques have been receiving a lot of attention recently, while addressing the problem of spatial upsampling. SR approaches can be categorized into reconstruction based SR [6], [10], [11], and example based SR [3], [8], [19]. While the former approach computes the high resolution images by simulating the image formation process, the latter aims at generating high resolution images based on small image segments extracted from training high resolution images. Example based approaches often reconstruct the high resolution image in a block-wise routine, hence all the operations are performed on blocks (patches) of pixels. Recently, example based techniques were improved further by introducing sparse coding [15] to the SR problem. Yang et al. [23], [24], [25] improved the approach presented in [3] by employing sparse representation paradigm and Dai et al. [4] and Kato et al. [12] made further contributions in the enhancement of the sparse representation based SR.

SR algorithms based on the sparse representation provide a high quality image reconstruction. However applying such methods to very high resolutions and extending the algorithms to videos are quite challenging due to the high complexity of the sparse coding. Block-wise operations in SR cause block artifacts at the block boundaries, and this issue is often solved by using overlapping blocks. Hence the number of block operations increases significantly and could reach to millions in the case of UHD resolution leading to an extremely slow image reconstruction. Furthermore, sparse coding methods have mainly focused on still images domain, thus the naïve frame-wise application of these approaches on videos can be extremely time consuming and impractical.

In this paper, we introduce an in-loop filtering mechanism adapted to the SR that replaces the exhaustive overlapping block operations, while retaining the high quality of the reconstructed image. Furthermore, we introduce a predictive approach for video SR, with which the high resolution blocks can be predicted using the previously super-resolved frames, reducing the computation time of the SR further.

II. Still image SR via sparse representation

The initial idea of sparse coding SR was proposed by Yang et al. [24], [25], where a low resolution frame is super-resolved to a higher resolution using a high resolution dictionary.

A. Single-frame SR

Let $\mathbf{D} = \{D_1, D_2, \ldots, D_k\} \in \mathbb{R}^{n \times k}$ be a high resolution *dictionary* with k different atoms of size n. Sparse coding enables representation of a vectorized block $x_t \in \mathbb{R}^n$ as

$$x_t = \mathbf{D}\alpha. \tag{1}$$

x_t represents the training high resolution data, and $\alpha \in \mathbb{R}^k$ represents the coefficients vector for sparse representation of x_t. The atoms of the dictionary are learned by the following minimization problem [12].

$$\underset{\{\mathbf{D},\alpha\}}{\text{minimize}} \left\|\mathbf{D}\alpha - x_t\right\|_2^2 + \eta \left\|\alpha\right\|_1, \tag{2}$$

where $\eta \geq 0$ represents the sparseness constraint strength. Solving this problem results in a high resolution dictionary.

SR based on the sparse representation also requires a low resolution dictionary $\mathbf{d} = \{d_1, d_2, \ldots, d_k\} \in \mathbb{R}^{\hat{n} \times k}$ for representing the low resolution input image, with $\hat{n}$ denoting the number of elements (pixels) in each low resolution atom. The low resolution dictionary can of course be modeled by degradation of the high resolution dictionary using downsampling and blurring operations. Low resolution dictionary can also be learned in parallel with the high resolution dictionary [23]. If $y_t \in \mathbb{R}^{\hat{n}}$ represents the vectorized low resolution training blocks (or features), then the following equation results in a

978-1-5090-3649-3/17/$31.00

low resolution dictionary, given that the high resolution and low resolution training data are paired.

$$\underset{\{\mathbf{d},\alpha\}}{\text{minimize}} \|\mathbf{d}\alpha - y_t\|_2^2 + \eta \|\alpha\|_1 \tag{3}$$

The same sparse coefficients α are used for the low resolution data according to Equation (3). This is a critical assumption building the core idea of the sparse coding. Once the atoms of the dictionaries are available, each low resolution block y can be modeled by the following sparse representation.

$$y = d_1\alpha_1 + d_2\alpha_2 + d_3\alpha_3 + \ldots, \tag{4}$$

and once the sparse coefficients are obtained, the high resolution version of the target block is created using the following.

$$x = D_1\alpha_1 + D_2\alpha_2 + D_3\alpha_3 + \ldots, \tag{5}$$

where x represents the high resolution block. As described earlier, this block-wise procedure generates artifacts and unwanted edges at the block boundaries. To avoid these artifacts, overlapping blocks are employed. As a simple approach, it is possible to average the feature values in overlapped regions between adjacent blocks [3], [12]. For combining neighboring high resolution blocks, two-dimensional Hanning windows can be applied to reduce the effect of overlapping.

B. Multi-frame SR

Kato et al. [12] and Dai et al. [4] presented approaches to utilize multiple low resolution frames for generating one target high resolution frame. In multi-frame SR, a set of low resolution frames Y_i, $i = 1, 2, \ldots, N$ (eg. consecutive frames of a video) are considered to be an outcome of different degradations of the high resolution target image X. Hence the correlation between the target low resolution frame Y and the auxiliary low resolution frames Y_i can result in a better sparse representation of the low resolution blocks. This approach can be very effective for video signals, since the adjacent frames of a video are highly correlated.

1) Registration via block matching: When using auxiliary frames, the target low resolution block should be registered to the co-located low resolution blocks in the auxiliary frames. Hence a reliable motion estimation is essential to proceed with this approach. Kato et al. [12] use a block matching method [1], [2] for registration of the target block to the auxiliary blocks. It is assumed that the positional relationship of blocks are fully described by parallel translation. In multi-frame SR, estimating the sub-pixel shifts between the images can enhance the results remarkably. There are several approaches for estimating positional shifts in sub-pixel accuracy. Shimizu and Okutomi [16] suggest a two-parameter simultaneous estimation method, which balances accuracy and complexity.

By the block matching method, the following similarity score is obtained between the target low resolution block y and an auxiliary block y_i.

$$sim(y, y_i) = \frac{y^\intercal y_i}{\|y\|_2 \cdot \|y_i\|_2} \in [0, 1] \tag{6}$$

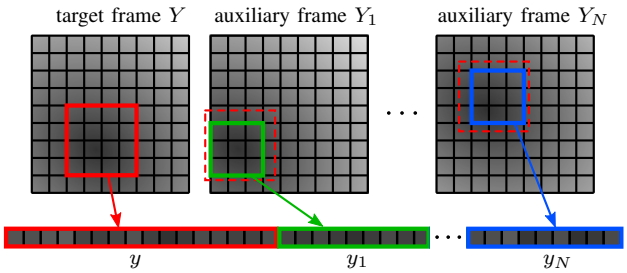


Fig. 1. Clipping and stacking of the auxiliary blocks.

Block matching is of course not the only registration scheme, and there exists other registration methods such as the optical flow estimation [5] adopted by Dai et al. [4].

2) Multi-frame sparse coding: Once the target low resolution block y is registered to the auxiliary blocks y_i, $i = 1, 2, \ldots, N$, where N is the total number of auxiliary frames containing a block with a similarity value above a certain threshold δ, all the vectorized blocks concatenate in one vector as

$$\tilde{y} = [y^\intercal \; y_1^\intercal \; y_2^\intercal \; \ldots \; y_N^\intercal]^\intercal, \text{with } \forall y_i : sim(y, y_i) > \delta. \tag{7}$$

This process is illustrated in Figure 1. Figure 1 also depicts the clipping operation, which extracts the pixels completely included in the cutoff line (solid boxes) to avoid boundary effects. The following optimization problem results in the required sparse coefficients for generation of the high resolution block using multi-frame low resolution blocks [12].

$$\underset{\{\alpha\}}{\text{minimize}} \left\|\tilde{\mathbf{d}}\alpha - \tilde{y}\right\|_2^2 + \eta \|\alpha\|_1, \tag{8}$$

where $\tilde{\mathbf{d}}$ represents the stacked low resolution dictionary created by means of the application of the downsampling and blurring operators on the deformed high resolution dictionary considering the registration of the auxiliary frames.

III. Proposed Video SR Approach

Inspired by the recent advances in video compression, including H.264/AVC [22] and H.265/HEVC [18], we present an efficient video SR approach, which is based on the core idea of the sparse coding adapted to the nature of videos.

A. Motion compensated block prediction in SR

The first step to extend the SR approaches to videos is to extend the upsampling units from a single frame to a set of frames, i.e. group of pictures (GOP). In our proposed scheme, every GOP contains an I-frame which is super-resolved individually using the multi-frame sparse coding approach, presented in Section II-B. However the rest of the frames in the GOP are P-frames, which have the possibility to be predicted partially by the previously super-resolved frames in the GOP. Table I demonstrates our proposed structure of a GOP. The frame order is different from the SR order. The size of a GOP is 9 and the frame in the center of the GOP is the I-frame. Every P-frame can exploit the information from all the previously super-resolved frames within the GOP and when

TABLE I
GOP STRUCTURE IN SR.

Frame type	P	P	P	P	I	P	P	P	P
Frame order	1	2	3	4	5	6	7	8	9
SR order	6	8	2	4	1	5	3	9	7

appropriate skip the sparse coding process and instead predict the high resolution blocks from the neighboring frames.

A crucial step in block prediction is motion estimation. We use the block matching approach described in Section II-B1 for motion estimation. The block matching information is available for every block of each frame within the GOP. By using the block matching information and examining the similarity scores (6) between the low resolution block y and its corresponding matched blocks y_i from the adjacent frames, we can find the most similar auxiliary block $y_{\hat{i}}$ amongst the blocks with a similarity value above the threshold δ, namely the *reference* block.

$$y_{\hat{i}} = \max_{\{i\}} sim(y, y_i),\ i \in \{1, 2, \ldots, N\}. \tag{9}$$

If the reference block $y_{\hat{i}}$ is extracted from an already super-resolved frame, then the current block y can skip the sparse coding procedure and be represented by the following.

$$y = y_{\hat{i}} + y_{res}, \tag{10}$$

where y_{res} is the *residual* block representing the disparity between y and $y_{\hat{i}}$. Given that the frame containing $y_{\hat{i}}$ is already upsampled and the position of the block is known, the high resolution version of the reference block $x_{\hat{i}}$ can be extracted. Hence the high resolution version of y will be as following.

$$x = x_{\hat{i}} + x_{res}, \tag{11}$$

where x_{res} is the upsampled version of y_{res}. Since the residual pixels are often close to zero, and they contain no significant details, we use a simple bi-cubic upsampling for obtaining x_{res}. The prediction procedure leads to a fast reconstruction of x by skipping the sparse coding process.

Block matching operates with a sub-pixel accuracy. To achieve a finer approximation and take advantage of the sub-pixel block matching, we employ a half-pixel interpolation [20] of the pixel space using bilinear filters, enabling a more accurate prediction of the original block.

B. In-loop filters with non-overlapping blocks

As discussed in Section II-A, blocking artifacts occur in SR due to the nature of block-wise operations. Overlapping blocks are employed conventionally as a solution. Although effective in terms of the reconstruction quality, introducing overlapping blocks increases the complexity of SR significantly. Blocking artifacts appear to be a very well known issue in video compression, too, and in-loop filtering is proved to be very effective in video coding in dealing with such artifacts. H.264/AVC and H.265/HEVC employ pixel-wise filtering operations [9], [13], [14] with non-overlapping blocks, which improve the visual quality of the reconstructed images. That inspired us to design in-loop filtering in the SR problem.

1) Deblocking filters: Deblocking filters operate at the pixel level and modify the pixel values at the block boundaries. Figure 2(b) illustrates a one-dimensional case of a blocking artifact. For such a scenario, a typical deblocking filter modifies the pixels in the vicinity of the boundary, eg. l_0 and c_0, so that a smoother transition takes place at the boundary.

In order to avoid high complexity operations and over-smoothening of the image, we impose the filters to one layer of pixels at the boundary. Let C denote the current block undergoing the SR by sparse coding. T and L represent the top and left blocks which are already super-resolved. The deblocking mechanism is performed between C and T (if T is available), then between C and L (if L is available). For the sake of simplicity, we use the one-dimensional notation presented in Figure 2(b) in the following formulations, as the procedure will be similar for the two-dimensional case.

A strong deblocking mechanism will activate when a block is predicted from a neighboring frame. A normal deblocking mechanism can also be triggered when the following holds.

$$|c_0 - l_0| > |c_0 - c_1| + |l_0 - l_1| \tag{12}$$

In case of a deblocking, c_0 will go through the followings.

$$c_0 = c_0 + (l_1 + 2l_0 - 6c_0 + 2c_1 + c_2)/8, \tag{13}$$

$$c_0 = c_0 + (-3l_1 + 9l_0 - 9c_0 + 3c_1)/16. \tag{14}$$

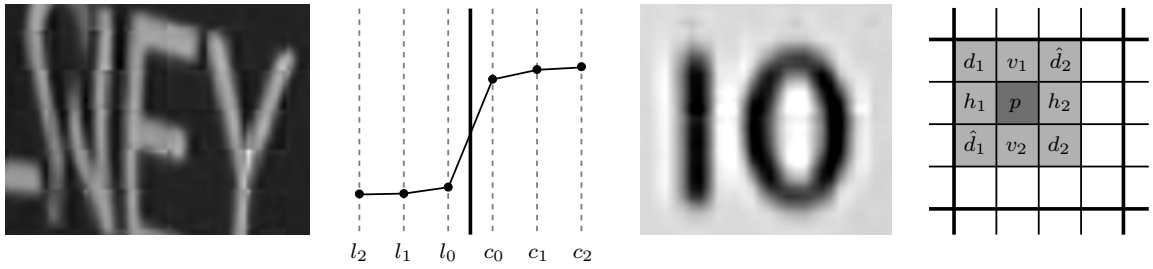
The modified value of l_0 is calculated by exchanging l and c in (13) and (14). Strong filtering is performed using (13), while normal filtering is performed using (14). The impulse responses of the (13) and (14) are $(1, 2, 2, 2, 1)/8$ and $(3, 7, 9, -3)/16$, respectively. The target pixel values remain unchanged in both cases in the presence of a perfect ramp. The above coefficients are extracted using heuristics.

Since the filtering is applied on a very narrow margin, the edges are retained in the reconstructed image. However ringing artifacts may appear near the sharp edges. Another in-loop process described next can eliminate these artifacts.

2) Adaptive pixel-wise operation: We propose an adaptive pixel-wise operation (APO), which can remove pixel-wise distortions and ringing artifacts as in Figure 2(c). This operation is also an in-loop (block-wise) process taking place directly after the deblocking filtering on every pixel of each block.

Figure 2(d) illustrates a 4×4 super-resolved and deblocked block going through the APO. All the pixels experience the APO and in the case of Figure 2(d), pixel p is about to undergo the process. A mask of 3×3 is created over each pixel (shaded pixels in Figure 2(d)) to monitor the variation of intensity values in horizontal, vertical and diagonal directions. The following conditions need to be checked for pixel p.

$$\begin{cases} p < i_1 \ \wedge \ p < i_2, & (1) \\ p > i_1 \ \wedge \ p > i_2, & (2) \\ (p < i_1 \wedge p = i_2) \ \vee \ (p = i_1 \wedge p < i_2), & (3) \\ (p > i_1 \wedge p = i_2) \ \vee \ (p = i_1 \wedge p > i_2), & (4) \end{cases} \tag{15}$$



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

Fig. 2. (a) An example of blocking artifacts in an image. (b) 1-D example of a block boundary with blocking artifact. (c) An example of an image with pixel distortions (eg. artifacts above the characters). (d) Pixel p to undergo the APO in a block of 4×4.

where $\wedge$ and $\vee$ represent the logical *and* and logical *or*, respectively, and $i \in \left\{h, v, d, \hat{d}\right\}$ (see Figure 2(d)).

The conditions in (15) represent four one-dimensional patterns in horizontal, vertical or diagonal directions depending on the value of i. (15.1) and (15.2) are associated with a local valley and a local peak along the selected one-dimensional pattern, and (15.3) and (15.4) represent concave and convex corners. The above conditions are associated with pixel distortions, and when any of them holds the APO will be activated and p will be updated to the mean value of i_1 and i_2 in the selected direction. If condition (15) is satisfied for more than one direction, p changes to

$$p = \frac{1}{K} \sum_{i \in \left\{h,v,d,\hat{d}\right\}} \left(\frac{i_1 + i_2}{2}\right), \tag{16}$$

where $K \in \{1, 2, 3, 4\}$ denotes the number of directions that can satisfy the conditions in (15).

IV. Experimental results

We performed a comprehensive analysis of our proposed approach in comparison with state of the art sparse representation based SR using an optimized C++ implementation of the models on an Intel(R) Core i7 3.20GHz machine with 16GB physical memory. We applied Yang et al. [23] bi-level dictionary learning, and Kato et al. [12] multi-frame SR. We trained a high resolution dictionary with 512 atoms of size 10×10 pixels and its associated low resolution dictionary with a scaling factor of 2. We employed 8 auxiliary low resolution frames (GOP of size 9) for the upsampling. It is worth noting that every type of multi-frame sparse representation based SR, and in general all the existing multi-frame overlapping block upsampling methods can be applied here, as the focus of our work is in extending the existing image SR approaches to the videos, and our approach does not change the nature of the SR, and it can be applicable to any block-wise algorithm.

The test data was 13 different sequences [17], [21] of each 1 GOP (9 frames), which covered different types of visual content. We performed an upsampling of 1080p HD to UHD. The low resolution data was obtained by downsampling the original UHD content using FFmpeg library [7]. We conducted tests using the sparse coding based SR on a frame by frame basis (all frames in the GOP were I-frames), while there was no overlap between the processed blocks. This was considered as the baseline system. We also tested under the

TABLE II
Quality and complexity evaluations of non-overlapping SR (baseline), overlapping SR (state of the art), non-overlapping SR enhanced by in-loop filters, and the predictive mechanism. Results correspond to an upsampling of 1920×1080 to 3840×2160.

Data	Baseline		State of the art		In-loop filters		In-loop + Prediction	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Book	40.24	0.9776	**41.00**	0.9812	**41.00**	**0.9816**	40.62	0.9813
Boxing	34.93	0.9443	35.69	0.9536	**35.99**	0.9573	35.71	**0.9575**
CalendarAndPlants	37.05	0.9727	37.79	0.9777	**37.87**	**0.9787**	37.82	**0.9787**
CampfireParty	34.38	0.9168	34.84	**0.9231**	**35.05**	0.9223	35.04	0.9223
Discus	30.13	0.9191	**30.81**	**0.9297**	30.73	0.9280	29.57	0.9142
Manege	27.14	0.8502	**27.76**	**0.8654**	27.63	0.8602	27.50	0.8566
Netball	37.14	0.9497	37.80	0.9557	**37.89**	**0.9571**	37.59	0.9570
ParkAndBuildings	30.68	0.9311	**31.28**	**0.9425**	31.16	0.9415	30.72	0.9388
Ribbon	40.96	0.9645	41.80	0.9682	**42.12**	**0.9696**	**42.12**	**0.9696**
RushHour	36.71	0.9274	37.44	0.9366	**37.67**	0.9388	37.62	**0.9390**
ShakeNDry	36.93	0.9076	37.47	0.9167	**37.77**	0.9188	37.76	**0.9196**
TallBuildings	30.23	0.8947	**30.75**	**0.9041**	30.60	0.8989	30.33	0.8971
TrafficFlow	34.42	0.8938	35.01	**0.9018**	**35.09**	0.9004	34.98	0.9002
Average processing time [sec]	10,092		24,506		10,523		9,042	
Super-resolved blocks per frame	82,944		230,400		82,944		∼69,605	

same conditions while there was a 40% overlap between the processed blocks to present the state of the art system. 40% overlap provides a favorable image reconstruction with respect to the baseline, and it is sufficient to remove the blocking artifacts. The two described systems are the references for our analysis.

We applied in-loop filters with non-overlapping blocks on an all I-frame configuration without any block prediction to analyze the effect of in-loop filters and compare their performance with the overlapping blocks mechanism. As the last step, we enabled 8 P-frames in the GOP and allowed prediction of the blocks from neighboring frames. This was tested while the in-loop filters were enabled.

Table II summarizes the objective evaluation of the proposed method in terms of quality and complexity in comparison with the two references. Introducing overlapping blocks (state of the art) enhances the image quality as expected, resulting in an average increase of 0.65 dB in PSNR, while increasing the number of upsampled blocks. Accordingly, the processing time has an 143% increase. Furthermore, the number of processed blocks increases by a factor of 3 compared to the baseline.

Applying in-loop filters on top of non-overlapping blocks, on the other hand, does not increase the number of processed blocks. Moreover, the complexity of the filters compared to the sparse coding routine is negligible and the processing times remain at the same level as the baseline. In terms of quality, in-loop filters provide an average of 0.74 dB improvement in PSNR and 0.008 improvement in SSIM over the baseline. Introducing the P-frames to the SR and enabling the block prediction from neighboring frames reduces the complexity further with respect to the all I-frame in-loop filtering case. In this scenario, the average number of super-resolved blocks decreases to 69,605 per frame and the processing time is 10% lower than the baseline, while providing an acceptable quality in terms of PSNR and SSIM.

Although PSNR and SSIM are widely used in evaluation of SR, subjective evaluation also plays an important role in incorporating the human perception in the analysis of different methods. Hence we also analyzed the quality of reconstructed high resolution images from a subjective perspective. Figure 3 shows different types of textures upsampled by the non-overlapping block based SR (baseline), overlapped block based SR (state of the art), and the proposed methods. The improvements over the baseline is vivid for both the state of the art and the proposed approach, and there is not a significant visual difference between the two. In-loop filters are able to remove the blocking artifacts and pixel-wise distortions successfully, while retaining the general consistency of some of the most challenging textures in Figure 3. Furthermore, block prediction does not degrade the quality of the reconstructed texture and keeps the general consistency of the image.

V. Conclusion

Our proposed sparse coding based video SR approach operates on a GOP basis, within which the frames have the possibility to be partially predicted using the previously upsampled frames. It reduces the computation time of the SR by skipping the expensive sparse coding process for the blocks that are highly correlated with the co-located blocks in the adjacent frames. We developed an adaptive in-loop filtering approach that reduces the blocking and pixel-wise artifacts in block based SR, while providing favorable objective and subjective quality in images. This approach promises an alternative to the exhaustive routine of overlapping blocks mechanism in the state of the art SR, which happens to be extremely complex. In-loop filters often provide better PSNR values to the overlapping blocks routine, while keeping the framework almost at the same complexity level as the non-overlapping blocks processing case. In total we reduced the complexity of the sparse coding approach by 63%, while providing the same level of image quality. The proposed video SR framework is an important step in extending the current image SR methods to videos and enabling to reach very high resolutions such as UHD.

Acknowledgment

The work described in this paper has been conducted within the project COGNITUS. This project has received funding from the European Unions Horizon 2020 research and innovation program under grant agreement No 687605.

References

[1] D. Barreto, L. D. Alvarez, and J. Abad. Motion Estimation Techniques in Super-Resolution Image Reconstruction. A Performance Evaluation. In *Virtual Observatory: Plate Content Digitization, Archive Mining and Image Sequence Processing*, pages 254–268, 2006.

[2] G. M. Callico, S. Lopez, O. Sosa, J. F. Lopez, and R. Sarmiento. Analysis of Fast Block Matching Motion Estimation Algorithms for Video Super-Resolution Systems. *IEEE Transactions on Consumer Electronics*, 54(3):1430–1438, 2008.

[3] H. Chang, D.-Y. Yeung, and Y. Xiong. Super-Resolution through Neighbor Embedding. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 275–282, 2004.

[4] Q. Dai, S. Yoo, A. Kappeler, and A. K. Katsaggelos. Dictionary-based Multiple Frame Video Super-resolution. In *IEEE International Conference on Image Processing (ICIP)*, pages 83–87, 2015.

[5] M. Drulea and S. Nedevschi. Total Variation Regularization of Local-global Optical Flow. In *IEEE International Conference on Intelligent Transportation Systems*, pages 318–323, 2011.

[6] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar. Fast and Robust Multiframe Super Resolution. *IEEE Transactions on Image Processing*, 13(10):1327–1344, 2004.

[7] FFmpeg team. FFmpeg. `https://www.ffmpeg.org`, 2000–2017.

[8] W. T. Freeman, T. R. Jones, and E. C. Pasztor. Example-Based Super-Resolution. *IEEE Computer Graphics and Applications*, 22(2):56–65, 2002.

[9] C.-M. Fu, E. Alshina, A. Alshin, Y.-W. Huang, C.-Y. Chen, C.-Y. Tsai, C.-W. Hsu, S. Lei, J.-H. Park, and W. Han. Sample Adaptive Offset in the HEVC Standard. *IEEE Trans. Circuits Syst. Video Techn.*, 22(12):1755–1764, 2012.

[10] R. C. Hardie, K. J. Barnard, and E. E. Armstrong. Joint MAP Registration and High-Resolution Image Estimation using a Sequence for Undersampled Images. *IEEE Transactions on Image Processing*, 6(12):1621–1633, 1997.

[11] A. Kanemura, S.-i. Maeda, and S. Ishii. Superresolution with Compound Markov Random Fields via the Variational EM Algorithm. *Neural Networks*, 22(7):1025–1034, 2009.

[12] T. Kato, H. Hino, and N. Murata. Multi-Frame Image Super Resolution Based on Sparse Coding. *Neural Networks*, 66:64–78, 2015.

[13] P. List, A. Joch, J. Lainema, G. Bjøntegaard, and M. Karczewicz. Adaptive Deblocking Filter. *IEEE Trans. Circuits Syst. Video Techn.*, 13(7):614–619, 2003.

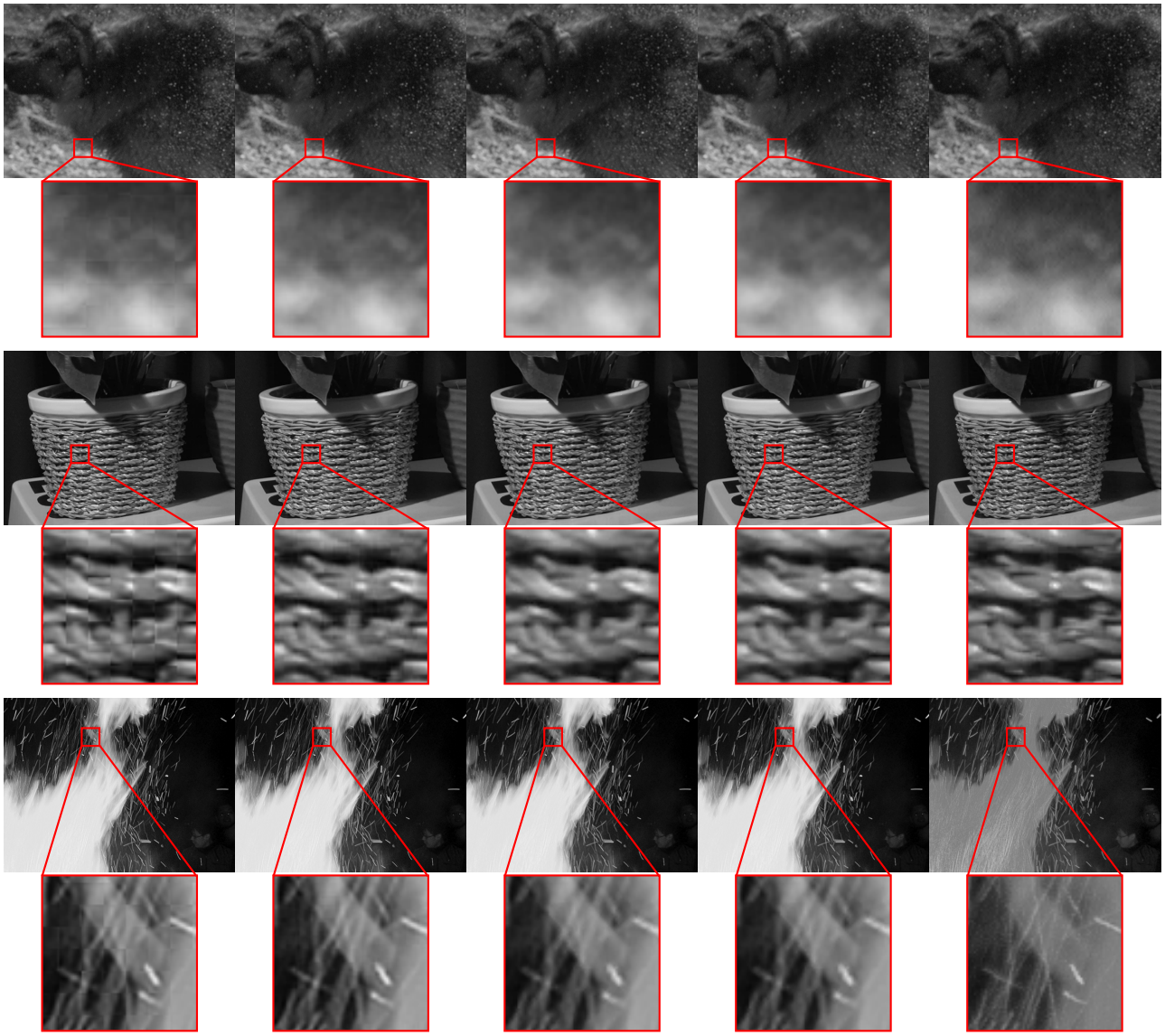


Fig. 3. Portions of *ShakeNDry*, *CalendarAndPlants* and *CampfireParty*. From left to right: Baseline, state of the art, in-loop filters, in-loop and predictive mechanism (some portions of the magnified areas are predicted), and the ground truth.

[14] A. Norkin, G. Bjøntegaard, A. Fuldseth, M. Narroschke, M. Ikeda, K. Andersson, M. Zhou, and G. V. D. Auwera. HEVC Deblocking Filter. *IEEE Trans. Circuits Syst. Video Techn.*, 22(12):1746–1754, 2012.

[15] B. Olshausen and D. Field. Emergence of Simple-Cell Receptive Field Properties by Learning a Sparse Code for Natural Images. *Nature*, 381:607–609, 1996.

[16] M. Shimizu and M. Okutomi. Multi-Parameter Simultaneous Estimation on Area-Based Matching. *International Journal of Computer Vision*, 67(3):327–342, 2006.

[17] L. Song, X. Tang, W. Zhang, X. Yang, and P. Xia. The SJTU 4K Video Sequence Dataset. In *Fifth International Workshop on Quality of Multimedia Experience (QoMEX2013)*, pages 34–35, 2013.

[18] G. J. Sullivan, J.-R. Ohm, W. Han, and T. Wiegand. Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1649–1668, 2012.

[19] J. Sun, N. Zheng, H. Tao, and H.-Y. Shum. Image Hallucination with Primal Sketch Priors. In *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 729–736, 2003.

[20] K. Ugur, A. Alshin, E. Alshina, F. Bossen, W. Han, J. Park, and J. Lainema. Motion Compensated Prediction and Interpolation Filter Design in H.265/HEVC. *J. Sel. Topics Signal Processing*, 7(6):946–956, 2013.

[21] R. Weerakkody, M. Naccari, and M. Mrak. UHD Test Sequences. Technical report, JCTVC-O0332, 2013.

[22] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the H.264/AVC Video Coding Standard. *IEEE Trans. Cir. and Sys. for Video Technol.*, 13:560–576, 2003.

[23] J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. Huang. Coupled Dictionary Training for Image Super-Resolution. *IEEE Transactions on Image Processing*, 21(8):3467–3478, 2012.

[24] J. Yang, J. Wright, T. Huang, and Y. Ma. Image Super-Resolution as Sparse Representation of Raw Image Patches. In *Proceedings of the 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[25] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image Super-Resolution via Sparse Representation. *IEEE Transactions on Imgage Processing*, 19(11):2861–2873, 2010.