# Joint Adaptive Sparsity and Low-Rankness on the Fly: an Online Tensor Reconstruction Scheme for Video Denoising (Supplementary Material)

Bihan Wen Yanjun Li Luke Pfister Yoram Bresler
Electrical and Computer Engineering and Coordinated Science Laboratory
University of Illinois at Urbana-Champaign, IL, USA.
{bwen3, yli145, lpfiste2, ybresler}@illinois.edu

## 1. More Denoising Results

We first illustrate an example of color video denoising, using a simple extension of the proposed SALT video denoising. Besides, we report both the denoised peak signal-to-noise ratio (PSNR) and the structural similarity (SSIM) [1], which are objective measures commonly used to assess video quality, obtained by the proposed SALT method over both ASU [2] and TUT [3] datasets, and comparing to state-of-the-art competing methods.

### 1.1. Extension to Color Video Denoising

This work focuses on proposing a novel and effective video denoising method, thus only the gray-scale video denoising algorithm is discussed in details. Prior work on online transform learning [4] provided a simple extension from gray-scale to color image denoising using transform learning. Similarly, here we extend our gray-scale SALT video denoising algorithm to color video denoising, by learning a 4D sparsifying transform for the extracted 4D tensors.

Figure 1 illustrates the visual comparison of the denoising results, by showing one frame of the denoised color *Stefan* at $\sigma = 20$ (the clean and noisy frames are shown in Fig. 1(a) and (b)), obtained by color extension of SALT (see Fig. 1(c)) and RGB color independent VBM3D (see Fig. 1(d)). The denoised frame by color SALT clearly preserves more local structures and details, while RGB color independent VBM3D generates blurry regions, *e.g.*, the zoomed-in region in the red box. It is also evident that the denoised frame y SALT exhibits much lower reconstruction error (see Fig. 1(e) and (f)).

Such simple extension to color video denoising using SALT may not be optimal. Besides, there are other possible extensions to the proposed SALT method. For example, this work demonstrates Gaussian noise removal, which has been extensively studied [4–10]. Similar approaches can be generalized to remove other types of noise. Moreover, the noise standard deviation is normally assumed to be known. Prior works [11, 12] provided approaches for noise level estimation, which can be combined with the proposed SALT method. We leave the similar extensions of SALT to future work.

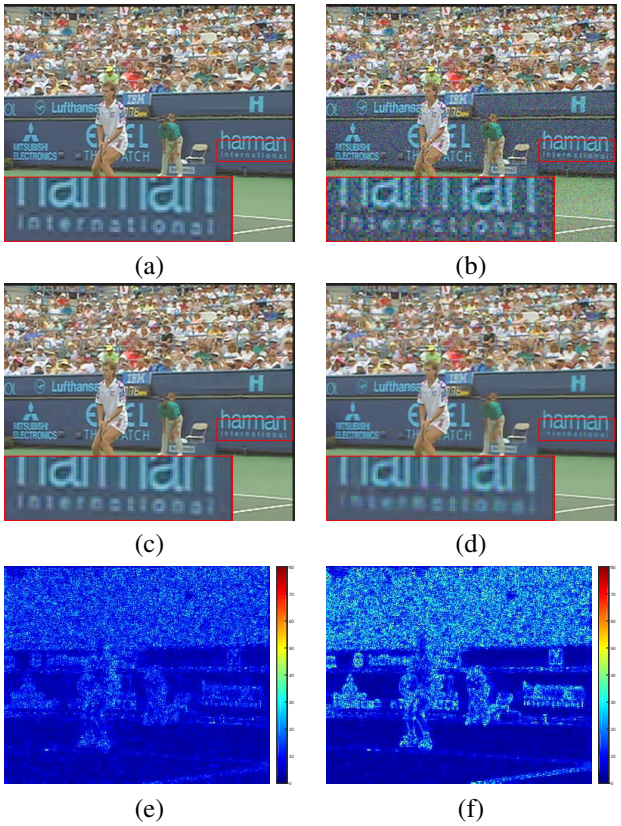

Figure 1. Denoising result: (a) One frame of the color video *Stefan*, (b) Frame corrupted with noise at $\sigma = 20$ (PSNR = 22.10 dB), (c) Denoised frame using the color video denoising extension of SALT (PSNR = 31.97 dB). (d) Denoised frame using single RGB channel VBM3D (PSNR = 28.76 dB). (e) Magnitude of error in (c). (f) Magnitude of error in (d).

| Methods | Local Sparse Model | | | Block Matching | Temporal Correlation | Non-local Method | |
|---|---|---|---|---|---|---|---|
| | Fixed | Adaptive | Online Update | | | Wienner Filtering | Low-Rank Approximation |
| fBM3D | ✓ | | | ✓ | | ✓ | |
| sKSVD | | ✓ | | | ✓ | | |
| VIDOSAT | | ✓ | ✓ | | ✓ | | |
| VBM3D / VBM4D | ✓ | | | ✓ | ✓ | ✓ | |
| BM-DCT | ✓ | | | ✓ | ✓ | | |
| BM-LR | | | | ✓ | ✓ | | ✓ |
| BM-TL | | ✓ | ✓ | ✓ | ✓ | | |
| **SALT** | | ✓ | ✓ | ✓ | ✓ | | ✓ |

Table 1. Comparison between the proposed SALT denoising and the competing methods.

### 1.2. Complete PSNR and SSIM tables

Table 2 and 3 list the denoised PSNR and SSIM values over the 28 videos in ASU dataset [2], and Table 4 and 5 list the denoised PSNR and SSIM values over the 8 videos in TUT dataset [3], which are obtained by our proposed SALT video denoising method, as well as VBM3D, VBM4D, and VIDOSAT, which are the major competing methods [1]. The SSIM values of the denoised results using SALT also show an average improvement over those using competing methods. SALT consistently provides the highest PSNRs for most videos and noise levels. There are only few cases in which SALT does not outperform VBM3D, but still provides comparable denoised results.

### 1.3. Denoised Video Generation

We generate the denoised video of *Gbicycle*, and both gray-scale and color *Stefan*, which are compressed using MPEG-4, using Matlab 2015b function *VideoWriter* on the Windows 10 operating system.

## 2. Selected Competing Methods

As discussed in the paper, there is a significant difference between denoising videos and other multi-frame image data, such as volumetric data or hyperspectral data. Thus, we restrict our investigation of competing denoising methods to those which are developed for video denoising and have publicly available implementations. Table 1 provides a more comprehensive comparison of the different attributes of the proposed SALT denoising and the selected competing methods. It would be interesting to extend the comparison to additional recent algorithms [13–15] when their implementations become publicly available.

Recent works have demonstrated that iterative processing schemes [6, 16, 17] can often enhance denoising algorithms and improve their average denoised PSNRs, subject to careful parameter tuning. For the sake of computational efficiency and fair comparison we did not apply any of the similar schemes in the SALT algorithm or the competing methods. We leave the study of combining iterative schemes with SALT denoising to future work.

[1] We observe that VBM3D generates denoised results with very low PSNR for several videos at $\sigma = 5$. We conjecture that there are some errors in VBM3D publicly available implementation.

## 3. Choice of Parameters

We provide a simple analysis and intuition to justify the choice of parameters used in our numerical experiments.

### 3.1. Thresholds $\rho$ and $\theta$

We choose $\rho = 3\sigma$ and $\theta = 1.1\sigma(\sqrt{K} + \sqrt{n_s})$ as the thresholds for sparse codes and singular values respectively. The following analysis is under the assumption that the clean vectorized and matricized tensors are exactly sparse under the adaptive transform and low-rank, respectively.

**Sparse code threshold** $\rho$: Suppose the clean vectorized tensor is $u = W^T\alpha \in \mathbb{R}^n$, the noisy observation is $\tilde{u} = u+e$, where $\alpha$ is a sparse vector and $e \sim N(0, \sigma^2 I_n)$ is additive Gaussian white noise, and the residual in the transform domain is $W\tilde{u} - \alpha = We \sim N(0, \sigma^2 I_n)$ and follows the same Gaussian distribution. In order to suppress the Gaussian noise in $W\tilde{u}$ and make $\hat{\alpha} = H_\rho(W\tilde{u})$ close to the true sparse code $\alpha$, one can choose a threshold $\rho$ that is proportional to $\sigma$. Empirically, we observe that $\rho = 3\sigma$ yields good final reconstruction results. Since the threshold is 3

standard deviations, the probability that a zero element in $\alpha$ becomes nonzero in $\hat{\alpha}$ is very low ($0.27\%$).

**Singular value threshold** $\theta$: Similarly, we assume that $U \in \mathbb{R}^{n_s \times K}$ has rank $r$ and its compact SVD is $U = \Lambda\Omega\Delta^T$ , where $\Lambda \in \mathbb{R}^{n_s \times r}$, $\Omega \in \mathbb{R}^{r \times r}$, and $\Delta \in \mathbb{R}^{K \times r}$. The noisy observation is $\tilde{U} = U + E$, where $E$ is additive Gaussian white noise of variance $\sigma^2$. When the noise level is low, the full SVD of $\tilde{U}$ is approximately

$$\tilde{U} \approx \begin{bmatrix} \Lambda & \tilde{\Lambda} \end{bmatrix} \begin{bmatrix} \tilde{\Omega}_1 & 0 \\ 0 & \tilde{\Omega}_2 \end{bmatrix} \begin{bmatrix} \Delta & \tilde{\Delta} \end{bmatrix}^T,$$

*i.e.*, the first $r$ singular vectors remain the same after adding the noise. Then the $(r+1)$-th singular value of $\tilde{U}$ is approximately $\|\tilde{\Lambda}^T\tilde{U}\tilde{\Delta}\| = \|\tilde{\Lambda}^T E\tilde{\Delta}\|$, where $\tilde{\Lambda}^T E\tilde{\Delta} \in \mathbb{R}^{(n_s-r)\times(K-r)}$ is also Gaussian white noise of variance $\sigma^2$. By a well-known result in random matrix theory, the above spectral norm is approximately $\sigma(\sqrt{n_s - r}+\sqrt{K-r})$ (see, *e.g.*, [18]). In practice, one can set the singular value threshold $\theta$ between the $r$-th and $(r+1)$-th singular value of $\tilde{U}$. Numerical experiments show that $\theta = 1.1\sigma(\sqrt{n_s} + \sqrt{K})$ achieves good denoising performance (note that the true rank $r$ is unknown).

### 3.2. Weights $\gamma_l$, $\gamma_s$, and $\gamma_f$

When we combine the sparse and low-rank approximations to form the final reconstruction, the weights are chosen such that the residual noise level is minimized.

Suppose we have two noisy copies $\tilde{u}_1, \tilde{u}_2$ of $u$, where $\tilde{u}_1 = u + e_1$, $\tilde{u}_2 = u + e_2$, and $e_1, e_2$ are Gaussian white noises of variances $\sigma_1^2$ and $\sigma_2^2$. Then the best convex combination of $\tilde{u}_1$ and $\tilde{u}_2$ that minimizes the residual noise variance is $(\sigma_2^2\tilde{u}_1 + \sigma_1^2\tilde{u}_2)/(\sigma_1^2 + \sigma_2^2)$, *i.e.*, the weights are inversely proportional to the variances.

Admittedly, the residual noise in the sparse and low-rank approximations are no longer Gaussian white noise. However, we find that choosing weights inversely proportional to the mean squared errors yields good reconstruction results. For example, the residual error in a sparse approximation has energy proportional to its sparsity level $s$. Therefore, the weight of sparse approximation is chosen as $\gamma_{s,i} = 60/s_i$, inversely proportional to sparsity level $s_i$. The choices of $\gamma_l$ and $\gamma_f$ are based on similar observations.

### 3.3. Other parameters

As the noise standard deviation increases from $\sigma = 5$ to $\sigma = 50$, we decrease the spatial search window size from $h = 30$ to $h = 16$ since running KNN over imperfect measurements within a larger window is more likely to cause mismatches, especially when the measurements are more corrupted (*i.e.*, with higher $\sigma$). Assuming the neighbor patches are similar, using a smaller window size is equivalent to performing a more regularized matching and reduces mismatches. We also use a larger tensor size $K$ for noisy videos with higher $\sigma$ so that each patch is denoised and aggregated more times to obtain better reconstruction result.

Though hyperparameter tuning is required, the proposed SALT denoising is an unsupervised approach, as it learns the transform model directly from the corrupted data, without training over a clean corpus. All the parameters are fixed for our denoising experiment after tuning, and SALT consistently outperforms all competitors in different datasets. Recent work proposed automatic parameter tuning for transform learning [19], which can be combined with the proposed SALT method for even better reconstruction performance.

## References


[1] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[2] P. Seeling and M. Reisslein, "Video traffic characteristics of modern encoding standards: H.264/AVC with SVC and MVC extensions and h.265/HVEC," *The Scientific World Journal*, vol. 2014, pp. 1–16, 2014.

[3] K. Dabov, A. Foi, and K. Egiazarian, "Video denoising by sparse 3d transform-domain collaborative filtering," in *Signal Processing Conference, 2007 15th European*, Sept 2007, pp. 145–149.

[4] S. Ravishankar, B. Wen, and Y. Bresler, "Online sparsifying transform learningpart i: Algorithms," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 625–636, 2015.

[5] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD : An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[6] Y. Romano and M. Elad, "Boosting of image denoising algorithms," *SIAM Journal on Imaging Sciences*, vol. 8, no. 2, pp. 1187–1219, 2015.

[7] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–45, Dec. 2006.

[8] S. Ravishankar and Y. Bresler, "Closed-form solutions within sparsifying transform learning," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 5378–5382.

[9] M. Maggioni, V. Katkovnik, K. Egiazarian, and A. Foi, "Nonlocal transform-domain filter for volumetric data denoising and reconstruction," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 119–133, 2013.

[10] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[11] D.-H. Shin, R.-H. Park, S. Yang, and J.-H. Jung, "Block-based noise estimation using adaptive gaussian filtering," *IEEE Transactions on Consumer Electronics*, vol. 51, no. 1, pp. 218–226, 2005.

[12] C. Liu, R. Szeliski, S. B. Kang, C. L. Zitnick, and W. T. Freeman, "Automatic estimation and removal of noise from a single image," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 2, pp. 299–314, 2008.

[13] A. Buades, J.-L. Lisani, and M. Miladinovic, "Patch-based video denoising with optical flow estimation," *IEEE Transactions on Image Processing*, vol. 25, no. 6, pp. 2573–2586, 2016.

[14] W. Dong, G. Li, G. Shi, X. Li, and Y. Ma, "Low-rank tensor approximation with laplacian scale mixture modeling for multiframe image denoising," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 12 2015, pp. 442–449.

[15] M. Rubinstein, C. Liu, P. Sand, F. Durand, and W. T. Freeman, "Motion denoising with application to time-lapse photography," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 313–320.

[16] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, "An iterative regularization method for total variation-based image restoration," *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 460–489, 2005.

[17] B. Wen, S. Ravishankar, and Y. Bresler, "Video denoising by online 3d sparsifying transform learning," in *IEEE International Conference on Image Processing (ICIP)*. IEEE, 2015, pp. 118–122.

[18] Y. Y. Z. Bai, "Limit of the smallest eigenvalue of a large dimensional sample covariance matrix," *The Annals of Probability*, vol. 21, no. 3, pp. 1275–1294, 1993.

[19] L. Pfister and Y. Bresler, "Automatic parameter tuning for image denoising with learned sparsifying transforms," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, 2017.

[20] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian, "Video denoising, deblocking, and enhancement through separable 4-d nonlocal spatiotemporal transforms," *IEEE Transactions on Image Processing*, vol. 21, no. 9, pp. 3952–3966, 2012.

| $\sigma$ | Akiyo (C) | | Akiyo (Q) | | Bus (C) | | Carphone (Q) | | Claire (Q) | | Coastguard (C) | | Coastguard (Q) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5 | 45.52 | 45.14 | 45.75 | 45.21 | 37.61 | 37.66 | 38.66 | 41.15 | 45.98 | 45.91 | 38.33 | 38.93 | 38.32 | 39.12 |
| | 45.67 | **46.29** | 45.80 | **46.50** | 37.85 | **39.38** | 40.90 | **42.54** | 46.17 | **47.11** | 39.38 | **39.79** | 39.59 | **40.19** |
| 10 | 42.57 | 41.98 | 42.08 | 41.15 | 33.39 | 33.40 | 37.36 | 37.43 | 42.58 | 42.09 | 34.62 | 35.14 | 34.82 | 35.37 |
| | 42.17 | **43.14** | 41.55 | **42.55** | 33.51 | **35.11** | 36.89 | **38.56** | 42.06 | **43.35** | 35.47 | **35.95** | 35.73 | **36.23** |
| 15 | 40.73 | 39.96 | 39.84 | 38.61 | 31.10 | 31.09 | 35.38 | 35.30 | 40.58 | 39.55 | 32.68 | 33.04 | 33.03 | 33.23 |
| | 39.92 | **41.01** | 38.98 | **40.13** | 31.17 | **32.62** | 34.66 | **36.29** | 39.64 | **40.99** | 33.30 | **33.85** | 33.57 | **34.15** |
| 20 | 39.23 | 38.39 | 37.95 | 36.65 | 29.58 | 29.53 | 34.05 | 33.83 | 39.00 | 37.51 | 31.37 | 31.61 | 31.73 | 31.71 |
| | 38.37 | **39.44** | 37.18 | **38.28** | 29.57 | **30.98** | 33.21 | **34.77** | 38.00 | **39.22** | 31.82 | **32.42** | 32.04 | **32.58** |
| 50 | 33.26 | 31.99 | 31.13 | 30.07 | 24.91 | 24.55 | 29.14 | 28.33 | 33.10 | 29.56 | 26.95 | 26.91 | 26.90 | 26.71 |
| | 32.81 | **33.46** | 31.51 | **32.36** | 25.05 | **26.16** | 28.93 | **30.00** | 32.59 | **33.14** | 27.41 | **28.04** | 27.53 | **28.07** |
| $\sigma$ | Container (C) | | Container (Q) | | Flower (C) | | Foreman (C) | | Foreman (Q) | | Grandma (Q) | | Hall (C) | |
| 5 | 42.05 | 42.18 | 43.05 | 42.87 | 38.31 | 38.10 | 39.83 | 39.98 | 35.26 | 40.37 | 43.56 | 43.63 | 41.66 | 41.49 |
| | 43.08 | **43.17** | 43.88 | **43.92** | 38.74 | **40.34** | 40.20 | **41.68** | 40.67 | **42.25** | 43.92 | **44.50** | 41.83 | **41.98** |
| 10 | 38.82 | 38.39 | 39.42 | 38.74 | 34.19 | 33.92 | 36.54 | 36.60 | 36.62 | 36.79 | 39.77 | 39.21 | 39.43 | 38.88 |
| | 39.37 | **39.57** | 39.65 | **39.87** | 34.31 | **35.87** | 36.58 | **37.96** | 36.68 | **38.11** | 39.64 | **40.33** | 39.27 | **39.76** |
| 15 | 36.87 | 36.22 | 37.31 | 36.49 | 32.00 | 31.66 | 34.72 | 34.72 | 34.65 | 34.70 | 37.50 | 36.40 | 38.06 | 37.08 |
| | 37.12 | **37.47** | 37.23 | **37.60** | 31.73 | **33.38** | 34.46 | **35.78** | 34.40 | **35.86** | 37.19 | **37.91** | 37.61 | **38.23** |
| 20 | 35.37 | 34.66 | 35.72 | 34.85 | 30.48 | 29.99 | 33.39 | 33.37 | 33.20 | 33.14 | 35.65 | 34.51 | 36.92 | 35.57 |
| | 35.57 | **35.83** | 35.58 | **35.82** | 29.89 | **31.43** | 32.95 | **34.27** | 32.75 | **34.20** | 35.60 | **36.14** | 36.34 | **36.99** |
| 50 | 29.80 | 28.70 | 29.34 | 28.23 | 24.27 | 23.32 | 28.79 | 28.39 | 28.05 | 27.72 | 30.26 | 28.69 | 31.32 | 29.33 |
| | 30.39 | **30.54** | 30.10 | **30.21** | 24.40 | **25.66** | 28.50 | **29.47** | 27.93 | **29.24** | 30.82 | **31.31** | 31.35 | **31.93** |
| $\sigma$ | Hall (Q) | | Mobile (C) | | Mobile (Q) | | Mother (C) | | Mother (Q) | | News (C) | | News (Q) | |
| 5 | 43.03 | 42.64 | 38.07 | 37.80 | 37.30 | 37.43 | 43.65 | 44.14 | 43.34 | 43.66 | 44.14 | 43.83 | 43.73 | 43.36 |
| | 43.10 | **43.33** | 39.41 | **40.08** | 39.40 | **39.84** | 44.17 | **45.05** | 43.82 | **44.96** | 44.24 | **44.91** | 43.71 | **44.35** |
| 10 | 40.05 | 39.15 | 34.17 | 33.72 | 33.37 | 33.43 | 40.58 | 40.76 | 39.80 | 39.62 | 40.82 | 40.29 | 39.81 | 39.14 |
| | 39.78 | **40.11** | 35.19 | **35.90** | 35.05 | **35.45** | 40.60 | **41.80** | 39.73 | **40.97** | 40.36 | **41.31** | 39.21 | **40.07** |
| 15 | **38.06** | 36.77 | 32.00 | 31.47 | 31.27 | 31.19 | 38.81 | 38.60 | 37.79 | 37.13 | 38.78 | 38.05 | 37.50 | 36.60 |
| | 37.52 | 38.05 | 32.70 | **33.55** | 32.49 | **32.96** | 38.41 | **39.59** | 37.43 | **38.61** | 37.97 | **38.98** | 36.60 | **37.59** |
| 20 | **36.42** | 34.88 | 30.41 | 29.81 | 29.73 | 29.55 | 37.47 | 37.07 | 36.27 | 35.41 | 37.19 | 36.37 | **35.76** | 34.75 |
| | 35.82 | 36.30 | 30.90 | **31.72** | 30.62 | **30.97** | 36.97 | **38.03** | 35.93 | **36.97** | 36.32 | **37.27** | 34.84 | 35.68 |
| 50 | 29.43 | 27.90 | 24.14 | 23.74 | 23.68 | 23.52 | **32.85** | 32.11 | 30.85 | 30.27 | 31.01 | 29.31 | 28.94 | 27.60 |
| | 30.10 | **30.60** | 25.31 | **26.08** | 25.12 | **25.35** | 32.25 | 32.72 | 31.09 | **31.84** | 30.90 | **31.52** | 29.30 | **29.88** |
| $\sigma$ | Salesman (Q) | | Silent (C) | | Silent (Q) | | Stefan (C) | | Suzie (Q) | | Tempete (C) | | Waterfall (C) | |
| 5 | 34.09 | 43.06 | 42.38 | 42.31 | 42.80 | 42.59 | 38.71 | 38.66 | 21.91 | 42.12 | 38.53 | 38.56 | 39.16 | 40.18 |
| | 43.43 | **43.85** | 42.43 | **43.13** | 42.95 | **43.54** | 38.93 | **40.32** | 42.04 | **43.57** | 39.28 | **40.24** | 41.28 | **42.12** |
| 10 | 39.24 | 38.59 | 38.64 | 38.16 | 38.73 | 38.23 | 34.61 | 34.55 | 38.09 | 38.67 | 34.84 | 34.74 | 35.63 | 36.49 |
| | 38.87 | **39.46** | 38.30 | **39.11** | 38.46 | **39.26** | 34.61 | **36.28** | 38.27 | **40.06** | 35.19 | **36.25** | 37.23 | **38.23** |
| 15 | 36.76 | 35.59 | 36.43 | 35.60 | 36.39 | 35.54 | 32.35 | 32.28 | 36.54 | 36.60 | 32.83 | 32.62 | 33.70 | 33.97 |
| | 36.22 | **36.90** | 35.98 | **36.81** | 35.91 | **36.82** | 32.09 | **33.81** | 36.21 | **37.94** | 32.89 | **33.98** | 34.81 | **35.75** |
| 20 | 34.81 | 33.50 | 34.73 | 33.85 | 34.63 | 33.64 | 30.73 | 30.66 | 35.30 | 35.18 | 31.39 | 31.10 | 32.24 | 32.13 |
| | 34.48 | **34.97** | 34.46 | **35.15** | 34.29 | **35.05** | 30.26 | **32.00** | 34.87 | **36.43** | 31.31 | **32.32** | 33.16 | **33.93** |
| 50 | 28.14 | 27.33 | 29.67 | 29.03 | 28.62 | 28.00 | 24.90 | 24.90 | 31.20 | 31.06 | 26.05 | 25.56 | 27.68 | 27.45 |
| | 29.27 | **29.75** | 29.90 | **30.53** | 29.25 | **29.99** | 24.88 | **26.17** | 30.58 | **31.60** | 26.64 | **27.33** | 28.54 | **29.11** |

Table 2. Comparison of video denoising PSNR values over ASU dataset. **Top Left**: VBM3D [3]; **Top right**: VBM4D [20]; **Bottom Left**: VIDOSAT [17];**Bottom right**: SALT video denoising. The QCIF (*i.e.* $144 \times 176$ resolution) and CIF (*i.e.* $288 \times 254$ resolution) videos are denoted with (Q) and (C), respectively. For each video and noise level, the best denoising PSNR is marked in bold.

| $\sigma$ | Akiyo (C) | | Akiyo (Q) | | Bus (C) | | Carphone (Q) | | Claire (Q) | | Coastguard (C) | | Coastguard (Q) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5 | 0.987 | 0.984 | 0.992 | 0.990 | 0.978 | 0.979 | 0.984 | 0.985 | 0.992 | 0.991 | 0.968 | 0.975 | 0.968 | 0.976 |
| | 0.987 | **0.988** | 0.992 | **0.993** | 0.979 | **0.986** | 0.984 | **0.988** | 0.991 | **0.993** | 0.976 | **0.978** | 0.978 | **0.980** |
| 10 | **0.979** | 0.974 | **0.985** | 0.978 | 0.944 | 0.947 | 0.969 | 0.969 | **0.987** | 0.982 | 0.926 | 0.940 | 0.932 | 0.944 |
| | 0.974 | **0.979** | 0.979 | **0.985** | 0.947 | **0.962** | 0.966 | **0.975** | 0.979 | 0.986 | 0.945 | **0.950** | 0.949 | **0.953** |
| 15 | **0.972** | 0.964 | **0.976** | 0.963 | 0.906 | 0.912 | 0.956 | 0.952 | **0.980** | 0.972 | 0.887 | 0.904 | 0.901 | 0.911 |
| | 0.959 | 0.968 | 0.963 | 0.974 | 0.914 | **0.933** | 0.946 | **0.960** | 0.964 | 0.976 | 0.912 | **0.921** | 0.920 | **0.928** |
| 20 | **0.963** | 0.953 | **0.963** | 0.945 | 0.871 | 0.879 | 0.943 | 0.936 | **0.972** | 0.960 | 0.852 | 0.870 | 0.870 | 0.876 |
| | 0.943 | 0.956 | 0.946 | 0.962 | 0.882 | **0.906** | 0.927 | **0.946** | 0.948 | 0.964 | 0.880 | **0.894** | 0.892 | **0.902** |
| 50 | **0.903** | 0.883 | **0.871** | 0.843 | 0.719 | 0.716 | **0.857** | 0.843 | **0.921** | 0.883 | 0.674 | 0.687 | 0.701 | 0.701 |
| | 0.821 | 0.839 | 0.818 | 0.861 | 0.733 | **0.771** | 0.811 | 0.848 | 0.831 | 0.853 | 0.718 | **0.751** | 0.741 | **0.768** |
| $\sigma$ | Container (C) | | Container (Q) | | Flower (C) | | Foreman (C) | | Foreman (Q) | | Grandma (Q) | | Hall (C) | |
| 5 | 0.973 | 0.974 | 0.982 | 0.979 | 0.989 | 0.989 | 0.972 | 0.974 | 0.981 | 0.983 | 0.987 | 0.986 | 0.971 | 0.969 |
| | **0.978** | **0.978** | **0.984** | 0.983 | 0.990 | **0.993** | 0.976 | **0.981** | 0.984 | **0.988** | 0.986 | **0.988** | **0.972** | **0.972** |
| 10 | 0.949 | 0.943 | 0.964 | 0.954 | 0.975 | 0.975 | 0.947 | 0.948 | 0.962 | 0.963 | 0.970 | 0.962 | 0.961 | 0.957 |
| | 0.954 | **0.956** | 0.963 | **0.966** | 0.975 | **0.982** | 0.952 | **0.960** | 0.964 | **0.972** | 0.966 | **0.972** | 0.959 | **0.962** |
| 15 | 0.929 | 0.917 | 0.949 | 0.937 | 0.961 | 0.960 | 0.927 | 0.928 | 0.946 | 0.945 | 0.949 | 0.927 | **0.954** | 0.947 |
| | 0.931 | **0.937** | 0.943 | **0.952** | 0.958 | **0.967** | 0.928 | **0.939** | 0.943 | **0.956** | 0.941 | **0.951** | 0.946 | 0.952 |
| 20 | 0.908 | 0.897 | 0.936 | 0.924 | 0.946 | 0.943 | 0.910 | 0.910 | 0.928 | 0.925 | 0.921 | 0.893 | **0.947** | 0.938 |
| | 0.909 | **0.917** | 0.926 | **0.938** | 0.939 | **0.950** | 0.905 | **0.919** | 0.920 | **0.938** | 0.916 | **0.929** | 0.933 | 0.942 |
| 50 | **0.824** | 0.808 | **0.861** | 0.841 | 0.830 | 0.829 | 0.811 | **0.813** | 0.818 | 0.818 | 0.795 | 0.762 | **0.886** | 0.870 |
| | 0.785 | 0.801 | 0.813 | 0.844 | 0.812 | **0.838** | 0.775 | 0.800 | 0.788 | **0.833** | 0.771 | **0.802** | 0.830 | 0.853 |
| $\sigma$ | Hall (Q) | | Mobile (C) | | Mobile (Q) | | Mother (C) | | Mother (Q) | | News (C) | | News (Q) | |
| 5 | 0.986 | 0.984 | 0.987 | 0.987 | 0.986 | 0.988 | 0.980 | 0.981 | 0.986 | 0.986 | 0.986 | 0.985 | 0.991 | 0.989 |
| | 0.986 | **0.986** | 0.989 | **0.992** | 0.991 | **0.993** | 0.982 | **0.984** | 0.986 | **0.990** | 0.987 | **0.988** | 0.990 | **0.992** |
| 10 | **0.979** | 0.974 | 0.971 | 0.970 | 0.967 | 0.970 | 0.964 | 0.964 | 0.969 | 0.967 | 0.977 | 0.973 | 0.981 | 0.976 |
| | 0.976 | **0.979** | 0.976 | **0.981** | 0.977 | **0.981** | 0.962 | **0.970** | 0.967 | **0.975** | 0.973 | **0.977** | 0.977 | **0.982** |
| 15 | **0.972** | 0.964 | 0.955 | 0.951 | 0.948 | 0.949 | 0.949 | 0.945 | 0.952 | 0.943 | **0.969** | 0.961 | **0.972** | 0.961 |
| | 0.963 | 0.971 | 0.959 | **0.969** | 0.960 | **0.966** | 0.940 | **0.953** | 0.944 | **0.958** | 0.958 | 0.964 | 0.961 | 0.970 |
| 20 | **0.964** | 0.952 | 0.935 | 0.928 | 0.928 | 0.924 | 0.933 | 0.927 | 0.934 | 0.919 | **0.959** | 0.949 | **0.961** | 0.946 |
| | 0.950 | 0.962 | 0.941 | **0.953** | 0.940 | **0.947** | 0.919 | **0.934** | 0.922 | **0.940** | 0.942 | 0.951 | 0.944 | 0.957 |
| 50 | **0.883** | 0.863 | 0.773 | 0.753 | 0.748 | 0.736 | **0.851** | 0.839 | 0.818 | 0.804 | **0.889** | 0.864 | **0.862** | 0.829 |
| | 0.846 | 0.882 | 0.813 | **0.842** | 0.813 | **0.824** | 0.783 | 0.802 | 0.785 | **0.824** | 0.831 | 0.840 | 0.829 | 0.857 |
| $\sigma$ | Salesman (Q) | | Silent (C) | | Silent (Q) | | Stefan (C) | | Suzie (Q) | | Tempete (C) | | Waterfall (C) | |
| 5 | 0.987 | 0.989 | 0.982 | 0.982 | 0.989 | 0.988 | 0.989 | 0.989 | 0.950 | 0.979 | 0.984 | 0.984 | 0.972 | 0.979 |
| | 0.990 | **0.991** | 0.983 | **0.985** | 0.989 | **0.991** | 0.988 | **0.991** | 0.980 | **0.984** | 0.985 | **0.989** | 0.983 | **0.986** |
| 10 | 0.976 | 0.970 | 0.961 | 0.954 | 0.974 | 0.968 | 0.978 | 0.978 | 0.954 | 0.958 | 0.966 | 0.965 | 0.935 | 0.948 |
| | 0.973 | **0.977** | 0.960 | **0.964** | 0.972 | **0.976** | 0.974 | **0.980** | 0.956 | **0.969** | 0.967 | **0.974** | 0.957 | **0.966** |
| 15 | 0.958 | 0.938 | 0.938 | 0.921 | 0.956 | 0.943 | **0.967** | 0.966 | 0.936 | 0.936 | 0.948 | 0.945 | 0.897 | 0.902 |
| | 0.951 | **0.958** | 0.934 | **0.942** | 0.951 | **0.959** | 0.958 | 0.965 | 0.932 | **0.951** | 0.946 | **0.956** | 0.924 | **0.938** |
| 20 | 0.931 | 0.901 | 0.911 | 0.890 | 0.934 | 0.915 | **0.955** | 0.952 | 0.918 | 0.916 | 0.928 | 0.923 | 0.854 | 0.850 |
| | 0.927 | **0.935** | 0.909 | **0.918** | 0.929 | **0.940** | 0.940 | 0.950 | 0.909 | **0.933** | 0.923 | **0.936** | 0.888 | **0.904** |
| 50 | 0.736 | 0.707 | 0.780 | 0.770 | 0.783 | 0.770 | **0.843** | 0.837 | **0.832** | 0.826 | 0.776 | 0.761 | 0.629 | 0.625 |
| | 0.779 | **0.803** | 0.767 | **0.794** | 0.793 | **0.826** | 0.804 | 0.826 | 0.773 | 0.808 | 0.787 | **0.807** | 0.694 | **0.731** |

Table 3. Comparison of video denoising SSIM values over ASU dataset. **Top Left**: VBM3D [3]; **Top right**: VBM4D [20]; **Bottom Left**: VIDOSAT [17];**Bottom right**: SALT video denoising. The QCIF (*i.e.* $144 \times 176$ resolution) and CIF (*i.e.* $288 \times 254$ resolution) videos are denoted with (Q) and (C), respectively. For each video and noise level, the best denoising SSIM is marked in bold.

| $\sigma$ | coastguard | | gbicycle | | gbus | | gflower | |
|---|---|---|---|---|---|---|---|---|
| 5 | 38.32 | 39.12 | 40.90 | 40.74 | 37.60 | 37.65 | 36.50 | 36.26 |
| | 39.60 | **40.25** | 39.91 | **42.08** | 37.84 | **39.43** | 36.84 | **38.13** |
| 10 | 34.81 | 35.35 | 37.64 | 37.33 | 33.37 | 33.39 | 32.11 | 31.81 |
| | 35.72 | **36.34** | 35.78 | **39.15** | 33.51 | **35.22** | 32.32 | **33.63** |
| 15 | 33.02 | 33.24 | 35.70 | 35.30 | 31.06 | 31.07 | 29.80 | 29.51 |
| | 33.58 | **34.29** | 33.38 | **37.11** | 31.16 | **32.79** | 29.71 | **31.15** |
| 20 | 31.71 | 31.71 | 34.19 | 33.77 | 29.53 | 29.52 | 28.19 | 27.94 |
| | 32.04 | **32.75** | 31.65 | **35.56** | 29.56 | **31.14** | 27.84 | **29.21** |
| 50 | 26.61 | 26.71 | 26.85 | 27.12 | 24.45 | 24.53 | 21.86 | 22.21 |
| | 27.53 | **27.90** | 26.57 | **29.81** | 25.03 | **26.38** | 22.27 | **23.55** |
| $\sigma$ | gforeman | | gmissa | | gsalesman | | gtennis | |
| 5 | 39.83 | 39.98 | 41.50 | 41.88 | 40.42 | 40.82 | 38.48 | 38.47 |
| | 40.20 | **41.78** | 42.30 | **42.43** | 41.55 | **41.60** | 38.28 | **39.45** |
| 10 | 36.54 | 36.60 | 39.62 | 39.84 | 37.27 | 37.12 | 34.67 | 34.44 |
| | 36.58 | **38.13** | 40.11 | **40.44** | 37.58 | **37.98** | 34.36 | **35.48** |
| 15 | 34.72 | 34.73 | 38.66 | 38.62 | 35.50 | 34.95 | 32.53 | 32.12 |
| | 34.46 | **36.08** | 38.60 | **39.09** | 35.22 | **36.02** | 32.19 | **33.29** |
| 20 | 33.36 | 33.37 | 37.87 | 37.73 | 34.04 | 33.29 | 31.03 | 30.61 |
| | 32.94 | **34.61** | 37.47 | **38.13** | 33.62 | **34.60** | 30.71 | **31.71** |
| 50 | 28.04 | 28.43 | 30.98 | 31.40 | 27.03 | 27.01 | 26.24 | 26.01 |
| | 28.49 | **29.77** | **33.34** | 33.32 | 28.76 | **29.57** | 26.31 | **27.44** |

Table 4. Comparison of video denoising PSNR values over TUT dataset. **Top Left**: VBM3D [3]; **Top right**: VBM4D [20]; **Bottom Left**: VIDOSAT [17];**Bottom right**: SALT video denoising. For each video and noise level, the best denoising PSNR is marked in bold.

| $\sigma$ | coastguard | | gbicycle | | gbus | | gflower | | gforeman | | gmissa | | gsalesman | | gtennis | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5 | 0.97 | 0.98 | 0.99 | 0.99 | 0.98 | 0.98 | 0.99 | 0.99 | 0.97 | 0.97 | 0.95 | 0.96 | 0.98 | 0.98 | 0.97 | 0.97 |
| | 0.98 | **0.98** | 0.99 | **0.99** | 0.98 | **0.99** | 0.99 | **0.99** | 0.98 | **0.98** | **0.96** | 0.96 | **0.99** | 0.98 | 0.97 | **0.97** |
| 10 | 0.93 | 0.94 | 0.98 | 0.98 | 0.94 | 0.95 | 0.97 | 0.97 | 0.95 | 0.95 | 0.94 | 0.94 | 0.96 | 0.96 | 0.92 | 0.92 |
| | 0.95 | **0.95** | 0.97 | **0.98** | 0.95 | **0.96** | 0.97 | **0.98** | 0.95 | **0.96** | **0.94** | 0.94 | **0.97** | 0.96 | 0.92 | **0.94** |
| 15 | 0.90 | 0.91 | **0.97** | 0.97 | 0.91 | 0.91 | 0.96 | 0.95 | 0.93 | 0.93 | 0.93 | 0.93 | 0.94 | 0.93 | 0.88 | 0.86 |
| | 0.92 | **0.93** | 0.95 | 0.97 | 0.91 | **0.94** | 0.96 | **0.97** | 0.93 | **0.94** | 0.93 | **0.93** | 0.94 | **0.95** | 0.88 | **0.90** |
| 20 | 0.87 | 0.88 | **0.96** | 0.96 | 0.87 | 0.88 | 0.94 | 0.94 | 0.91 | 0.91 | **0.92** | 0.92 | 0.92 | 0.90 | 0.83 | 0.82 |
| | 0.89 | **0.90** | 0.93 | 0.96 | 0.88 | **0.91** | 0.94 | **0.95** | 0.90 | **0.92** | 0.91 | 0.91 | 0.92 | **0.93** | 0.85 | **0.85** |
| 50 | 0.69 | 0.70 | 0.85 | **0.86** | 0.71 | 0.72 | 0.80 | 0.80 | 0.81 | **0.81** | **0.88** | 0.86 | 0.74 | 0.73 | 0.68 | 0.68 |
| | 0.74 | **0.75** | 0.80 | 0.85 | 0.73 | **0.78** | 0.81 | **0.84** | 0.77 | 0.80 | 0.80 | 0.78 | 0.78 | **0.80** | 0.69 | **0.69** |

Table 5. Comparison of video denoising SSIM values over TUT dataset. **Top Left**: VBM3D [3]; **Top right**: VBM4D [20]; **Bottom Left**: VIDOSAT [17];**Bottom right**: SALT video denoising. For each video and noise level, the best denoising PSNR is marked in bold.