Supplementary Material for

Efficient Learning of Image Super-resolution and Compression Artifact Removal with Semi-local Gaussian Processes

Younghee Kwon Kwang In Kim James Tompkin Jin Hyung Kim Christian Theobalt

This supplemental material consists of two sections. Section 1 describes details of experiments discussed in the main paper while Section 2 presents additional experimental results. Results from the main paper are computed over two databases, *DB1* and *DB2*. Samples from these are demonstrated in Figure 1.

1 Experimental details

***Easy* data points.** Figure 2 shows examples of data points which have small $P1$ values (see Sec. 3.3 of the main paper). These data points mostly lie at the major edges which show clean and strong change of pixel values and do not contain complex textures. For those patterns, the desired output should be less uncertain given the input. This supports using an adaptive noise model (Eq. 17 of the main paper).

Execution times. The run-times of the algorithms compared in the current paper are summarized in Table 1.

Details of the face image super-resolution experiments. For face image super-resolution experiments, we used FERET database [12]. The experimental scenario is that each input image contains a face which is roughly of size 25×25. Then, the objective is to super-resolve the facial part of the image 4 times along each dimension such that the resulting super-resolved face image is of size 100×100.

For this task, we exploited the fact that when the face images are well-aligned, each pixel of interest mostly shows (part of) specific objects. For instance, at the left-top corner of the face image, the regressor can focus on reconstructing the left eye and disregard e.g. the mouth. To facilitate this scenario, we performed automatic face detection and facial component detection on the database using the algorithms of [7] and [1], respectively. This produced 705 face images (see [1] for details). Based on the detected eye corner locations, the input image is scaled such that the x-coordinate value difference between the outer corners of two eyes becomes 50 (Fig. 3). Then, the detected face image is aligned into a 100×100-sized grid such that the outer corner of the left eye is located at $(25, 40)$ in this grid. In this way, the face images are roughly scale-normalized and aligned.

For each pixel location in a face image, we construct a GP regressor which is trained for reconstructing the images at that specific location. Since in general the facial component detection is erroneous (especially for low-resolution input faces), we explicitly made the regressor tolerant against small detection misplacements by sampling the training patch for the (i, j)-th regressor not only at (i, j) but also at its eight spatial neighbors. The parameters for regressors are adopted from the GP which was trained for the generic image database.

The experiments were performed in leave-one-person-out manner: For each face-containing test image, the images containing the same person of interest are removed from the training set and the regressors are subsequently trained. The low-resolution image is firstly 4-times-magnified along each dimension based on bicubic resampling on which the face detector and facial component detector is applied and the resulting face location is cropped, re-scaled, and aligned in the same way as for the training data. The detection of facial component in low-resolution interpolated image is facilitated by exploiting a strong prior on the joint location of several facial components [1]. Accordingly, even though we are using only the outer corner locations of eyes, we detect jointly the inner corners of eyes, mouth corners, and the nose. For quantitative evaluation, we prepared high-resolution ground truth images which are re-sized and aligned based on the object locations identified at low-resolution input images. As shown in Fig. 6 of the main paper, the average PSNR and SSIM improvements in our face-specific system are 2.10dB and 0.054, respectively while the corresponding values of the generic system [8] are 1.18dB and 0.032, respectively.

Effect of varying regression parameters. For rigorous evaluation, we tune the hyper-parameters of our algorithm based on a separate validation set. However, in practice, it may be that the user tries different parameter

Figure 1: *Top:* DB1, referred to in text by raster order. *Bottom:* DB2, sample from a 500-image personal photo collection.

Figure 2: Color map of *easy* data points in super-resolution experiments (see Sec. 3.3 of the main paper for details). For each pixel in the image, the corresponding image patch which is centered at the pixel of interest is compared with training patches and the corresponding $P1$ value is calculated. The pixels marked with green color corresponds to the data points whose $P1$ value is smaller than 0.005 (cf. Fig. 2 of the main paper). Flat image patches are pre-excluded.

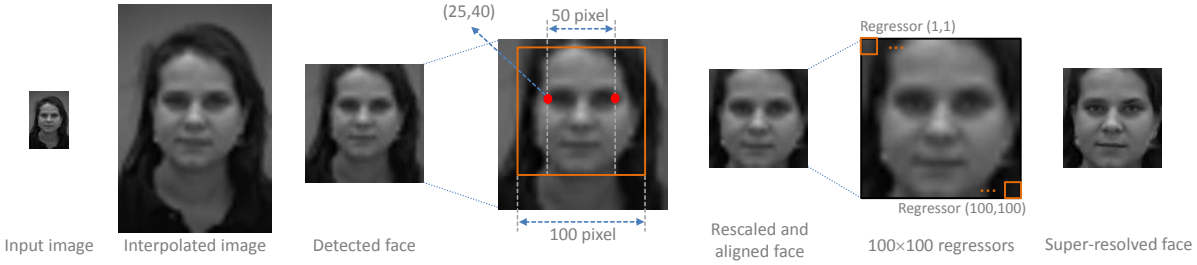

Figure 3: Schematic diagram of face image super-resolution.

Table 1: Average execution times of algorithms compared in the experiments; for a 512×512-sized image on a 3.6GHz machine.

Algorithm	Details	Execution time (sec.)
Ours	JPEG artifact removal	180
Nosratinia [10]	JPEG artifact removal	3
Gehler and Welling [5]	JPEG artifact removal	14
Roth and Black [13]	JPEG artifact removal	$2,600$
Foi *et al*. [3]	JPEG artifact removal	27
Laparra *et al*. [9]	JPEG (time for 256×256 image, $\times 4$)	280
Ours	Super-resolution; mag. factor 2	120
Freeman *et al*. [4]	Super-resolution; mag. factor 2	21
He and Siu [6]	Super-resolution; mag. factor 2	272
Chang *et al*. [2]	Super-resolution; mag. factor 3	28
Yang *et al*. [15]	Super-resolution; mag. factor 3	151
Kim and Kwon [8]	Super-resolution; mag. factor 3	60

Table 2: Impact of varying hyper-parameters on JPEG artifact removal performance: Average PSNR variation from the original results reported in the main paper (Q2).

$(\sigma^2)' \backslash b'$	$b \times 10$	b	$b/10$
1	-0.1342	-0.0036	-0.0197
σ^2	-0.0451	0	-0.1193
1^{-10}	-0.0559	-0.0045	-0.1194

combinations, visually inspects the results, and chooses the best result among them. To gain an insight into the practical utility of our algorithm in this scenario, we performed experiments with different combinations of regression parameters. Table 2 and Fig. 4 show the results of processing JPEG images. In the table, $(\sigma^2)'$ and b' are the new hyper-parameters while σ^2 and b are the original (automatically optimized) parameters. In preliminary experiments, we observed that changing b' and $(\sigma^2)'$ up to the orders of one and hundred from the originals do not introduce any significant differences in the visual quality and in the PSNRs of the results. This indicates that in practice, one could generate reasonable results with only few trials of parameter combinations (with relatively large search interval). For a general rule of thumb to choose parameters in this scenario, in all applications considered in this paper, we found that if we choose the initial guess of $(\sigma^2)'$ as the inverse of the squared mean distance of 10,000 randomly selected points, it does not deviate from the optimal σ^2 more than an order of magnitude.

The effect of varying hyper-parameters on the final results can be better understood when it is noted that in general, both b' and $(\sigma^2)'$ control the strength of regularization. For the case of images, strong regularization results in smoothing: assigning a large value to $(\sigma^2)'$ and a small value b' correspond to a strong regularization. When the $(\sigma^2)'$ and b' are much larger than σ^2 and smaller than b, respectively, the results become blurry and as expected (upper-right corner of Fig. 4). Interestingly, we obtained similar results from the weak regularization (see lower-left corner of Fig. 4). This can be attributed to the additional regularization effect of the combination step (see Eq. 4 of the main paper). While the individual reconstructed patches are much sharper (as each of them is closer to one of the stored examples), they are not as consistent in the overlap as the results obtained with strong regularization. As such, when they are combined, the final results are blurry.

2 Additional Results

This section presents additional experimental results which include example images and quantitative performance comparisons by PSNR and SSIM. We also present an example application of our algorithm to JPEG 2000 image enhancement where the objective is to remove ringing artifacts typically appearing in the encoded images. For

Figure 4: . Example of JPEG artifact removal with varying hyper-parameters. From top to bottom: the value of $(\sigma^2)'$ equals to 1, σ^2, and 1^{-10}, respectively. Please refer to the electronic version of this document for better visualization.

encoding JPEG 2000 images, we used the Kakadu software.[1] The preprocessing steps for JPEG 2000 image enhancement are the same to the JPEG case except for the fact that re-application of JPEG step is removed as JPEG 2000 images are free of block artifacts.

For comparison with existing JPEG 2000 image enhancement algorithms, Nosratinia's JPEG 2000 enhancement algorithm [11], FOEs, and PoEdges were evaluated and the reported results of Zhai *et al.*'s block-shift filtering-based approach [16] and Wang and Zhai's trilateral filters [14] are compared in Table 4.[2]

For the enhancement of JPEG and JPEG 2000 images, we present results corresponding to the Q3 compression quality of Table 2 in [10] (see main paper) and 0.1 and 0.15 bits per pixels (BPPs), respectively (Figs. 10-12). We also present a comparison with Laparra *et al.*'s support vector regression (SVR)-based algorithm [9] for JPEG artifact removal (Figs. 7 and 9). To enable a fair comparison with this algorithm, only six gray level images (from the 7-th to 12-th images in DB1; Fig. 4 of the corresponding paper) were used. These images were cropped to 256×256-pixels around the center and were encoded using Matlab's '*imwrite*' function with compression quality 7 (as was done in [9]).[3] In this set of experiments, the hyper-parameters of the proposed method optimized for Q2 (cf. main paper and Table 2 of [10]) were re-used as suggested by our on-line scenario.

For single-image super-resolution, results for magnification factors 3 and 4 along each dimension are presented in Figs. 14-16. Here, we additionally present a comparison with Yang *et al.*'s algorithm [15][4].

[1] `http://www.kakadusoftware.com/`.

[2] Here we also display the results corresponding to the case of compression ratio 0.1BPP for JPEG 2000. The readers are advised to be cautious regarding the absolute values of PSNR: even with the same bit rates and using the same encoding software, the PSNRs of the JPEG 2000-encoded images used in the current paper are slightly different (± 0.05dB) from those reported in [16, 14].

[3] This experimental setting is not the fundamental limitation of Laparra *et al.*'s algorithm. This setting enables us to directly use the code kindly shared by the authors at `http://www.uv.es/vista/vistavalencia/denoising_SVR/index.html`.

[4] We used the code and the corresponding dictionary (i.e., a set of basis vectors used in representing the images, trained for a magnification factor of 3, see [15] for details) kindly made available by the authors at `http://www.ifp.illinois.edu/~jyang29/`.

Table 3: Performance of different image enhancement algorithms for super-resolution (Mag. factor 2) and JPEG artifact removal (Q2) for DB1: increases of PSNRs (dB; top) and SSIMs ($\times 10^{-2}$; bottom) from JPEG images and bicubic-resampled images. Re-application of JPEG [10], SADCT [3], FOE [13], PoEdges [5], Chang *et al*. [2], Freeman *et al*. [4], He and Siu [6], and Kim and Kwon [8]. The best results are marked with bold letters.

Image	PSNR increase — JPEG						PSNR increase — Super-resolution				
	[10]	[13]	[10]+[13]	[5]	[3]	Ours	[2]	[4]	[6]	[8]	Ours
1	0.70	0.78	0.72	0.81	0.77	**0.90**	−1.77	−0.07	−0.13	1.36	**1.57**
2	0.64	0.61	0.65	0.78	0.93	**1.23**	−0.52	0.52	−0.36	2.45	**2.52**
3	0.57	0.59	0.58	0.60	0.68	**0.89**	−0.94	0.09	−0.27	1.48	**1.55**
4	0.49	0.55	0.50	0.61	0.62	**0.65**	−1.03	−0.21	−0.33	1.47	**1.48**
5	0.80	0.73	0.81	0.74	0.99	**1.15**	−1.35	−0.91	0.26	1.53	**1.57**
6	0.85	0.96	0.88	1.07	1.15	**1.18**	−1.60	0.40	−0.39	2.23	**2.26**
7	0.88	0.84	0.90	0.83	0.97	**1.06**	−1.37	−0.38	0.03	1.43	**1.47**
8	0.77	0.76	0.79	0.74	0.80	**0.91**	−1.43	−0.50	−0.03	1.17	**1.18**
9	1.21	1.15	1.25	1.35	1.49	**1.60**	−0.98	−0.33	−0.16	2.07	**2.17**
10	1.12	1.23	1.16	0.99	1.56	**1.69**	−2.41	1.11	0.34	**1.88**	**1.88**
11	0.87	0.83	0.90	0.82	0.97	**1.01**	−2.31	−0.34	−0.44	1.35	**1.38**
12	0.52	0.52	0.53	0.55	0.60	**0.64**	−1.41	−0.35	−0.21	**1.13**	1.09
13	0.68	0.67	0.70	0.70	**0.98**	0.97	−1.84	0.30	0.07	1.27	**1.36**
14	0.54	0.51	0.54	0.68	0.74	**0.89**	−2.43	0.02	0.08	1.89	**2.00**
15	0.88	0.96	0.91	1.01	1.34	**1.49**	−2.63	0.65	0.44	2.82	**3.02**
16	0.58	0.59	0.59	0.64	0.89	**1.15**	−2.28	0.52	−0.12	2.25	**2.35**

Image	SSIM increase — JPEG						SSIM increase — Super-resolution				
	[10]	[13]	[10]+[13]	[5]	[3]	Ours	[2]	[4]	[6]	[8]	Ours
1	2.34	2.16	2.36	**2.53**	1.80	2.49	−4.02	0.46	−1.10	**1.14**	1.12
2	0.66	0.96	0.64	1.01	0.98	**1.29**	−4.73	1.34	−2.08	**2.18**	2.16
3	1.17	0.89	1.13	0.99	0.22	**1.46**	−6.61	1.31	−0.83	**2.09**	2.06
4	1.44	1.07	1.40	1.49	0.78	**1.60**	−5.72	0.90	−1.64	**1.78**	1.75
5	2.25	**2.44**	2.30	2.38	2.33	2.41	−6.55	1.09	−0.06	1.79	**1.80**
6	2.82	3.23	2.92	**3.48**	3.15	2.94	−1.85	0.16	−0.55	**0.43**	**0.43**
7	1.56	1.30	1.56	1.44	1.32	**1.73**	−5.29	0.78	−0.03	**1.33**	1.31
8	1.77	1.18	1.75	1.43	1.04	**1.88**	−5.03	0.59	−0.36	**1.16**	1.13
9	2.49	2.33	2.54	2.70	2.55	**2.71**	−2.14	0.20	−0.68	**0.54**	**0.54**
10	2.68	2.95	2.78	2.73	**3.04**	2.86	−2.86	0.11	−0.43	**0.40**	0.39
11	1.72	1.72	1.76	1.66	1.59	**1.80**	−3.98	0.25	−0.59	**0.74**	0.73
12	0.74	0.28	0.67	0.72	0.23	**0.79**	−7.25	1.11	−1.10	**1.97**	1.93
13	2.52	1.49	2.55	2.02	**3.24**	3.23	−8.52	1.06	0.47	4.43	**4.52**
14	2.10	0.79	2.00	1.60	0.91	**2.69**	−11.08	0.90	0.37	5.24	**5.34**
15	3.58	3.45	3.68	3.25	4.09	**4.44**	−5.90	0.10	−0.23	2.47	**2.58**
16	3.18	3.23	3.25	2.76	3.38	**4.23**	−7.93	1.08	−0.02	4.46	**4.65**

Along with each example, the corresponding ground truth image is displayed. It should be noted that in general, in single-image super-resolution, there is no ground truth *image* even though often there is a ground truth *scene* which could be regarded as an image with an infinite resolution.[5] For the experiments, the ground truth image for each low-resolution image was generated by simulating the down sampling process (see main paper). The existence of ground truth images facilitates the numerical evaluation of each algorithm. However, it is not a requirement for the application of the proposed method. We suggest viewing the electronic version of this document for a clearer visualization of the results.

[5]Even this is not always true: e.g., synthesized images might not have an underlying ground truth scene.

Table 4: Performances of different JPEG 2000 enhancement algorithms for Lena, pepper, and bridge images (ΔPSNR(dB)).

Image	[16]	[14]	Our method
Lena (0.1BPP)	0.03	0.49	**0.50**
pepper (0.1BPP)	0.31	0.11	**0.47**
bridge (0.1BPP)	−0.05	N.A.	**0.04**
Lena (0.15BPP)	−0.34	N.A.	**0.54**
pepper (0.15BPP)	0.04	N.A.	**0.62**
bridge (0.15BPP)	**0.13**	N.A.	0.10

2.1 Enhancement of JPEG and JPEG 2000 images

The SVR-based method [9] successfully removed block artifacts and produced sharp edges (the last two columns of Fig. 7). However, it tended to leave ringing artifacts and accordingly, made the results more noisy than other methods. The results of our method are almost as sharp as the results of SVR at the edges. Furthermore, our results show fewer ringing artifacts as our method coherently reconstructs sharp edges and texture details. This is clearly visible in the visor of Lena (Figs. 5 and 7) and in the helmet of the biker in Fig. 6. The shape-adaptive DCT (SADCT) demonstrated comparable performance to the proposed method in terms of SSIM. However, detailed visual inspection (e.g. in the stripe pattern of the parrot in Fig. 6 and at the eyebrow of the woman of Fig. 6 in the main paper) reveals that the results of the proposed method are much cleaner (with fewer ringing artifacts) and contain more detail. Uncropped copies of Figure 6 from the main paper are included in Figure 18.

Nosratinia's algorithm successfully suppressed ringing artifacts in JPEG 2000 compressed images but it produced blurry images. Both FOEs and PoEdges have an advantage of being applicable to both JPEG and JPEG 2000 images even without relying on the training data. However, the superior performance of the proposed method over these algorithms (which is especially noticeable in the first column of Fig. 10 and the last two columns of Fig. 6) demonstrates that exploiting the knowledge of degradation process is desirable. Quantitative evaluation confirms this observation as plotted in Figs. 9 and 13 and shown in Tables 3-5.

2.2 Super-resolution

All tested super-resolution algorithms produced significantly sharper images than bicubic resampling. Yang *et al.*'s algorithm [15] and Freeman *et al.*'s algorithm [4] produced sharper but partially noisy images. The results of Chang *et al.*'s algorithm [2] are a little blurry especially at textured regions.

The proposed method and Kim and Kwon's algorithm [8] produced results as sharp as the results of [4] and [15], but with less noise. Thus, both algorithms improve both the visual quality and the PSNR and SSIM values (Fig. 13; our algorithm gives slightly better PSNR and SSIM values than [8]). However, for the same level of testing time and performance, our method is around 300 times faster in training. This can results in a significant difference in the utilities of the two algorithms: Our algorithm can be quickly adapted to new data (see examples of face image and document super-resolution in the main paper), which is not possible for [8].

Furthermore, we have 4 hyper-parameters to be tuned for each application (M, σ_P, σ^2, and b)[6] while in [8], 8 hyper-parameters need to be tuned (for each application). Since the kernel ridge regression used in [8] corresponds to the mean estimation of a GP, the two algorithms share the same set of hyper-parameters in the regression stage. The difference in the number of hyper-parameters stems from the fact that in [8] an explicit post-processing step is used to suppresses the ringing artifacts, which has additional hyper-parameters (to be tuned for each application). On the other hand, in our algorithm, the ringing artifact is suppressed by adaptively controlling the degree of regularization where the hyper-parameters can be shared in different applications. The time required for training and the number of hyper-parameters of [8] restricts the application domain significantly. For instance, the algorithm of [8] cannot be applied in interactive setting, i.e., the user tunes the parameters by visually inspecting results, which is feasible in our algorithm. Uncropped copies of Figure 4 from the main paper are included in Figure 17.

[6]The remaining hyper-parameters are fixed throughout the entire set of experiments in the current paper. See Table 1 of the main paper.

Figure 5: Examples of JPEG artifact suppression (Q3): (a) input JPEG images, (b) re-application of JPEG [10], (c) SADCT [3], (d) FOE [13], (e) PoEdges [5], and (f) our method.

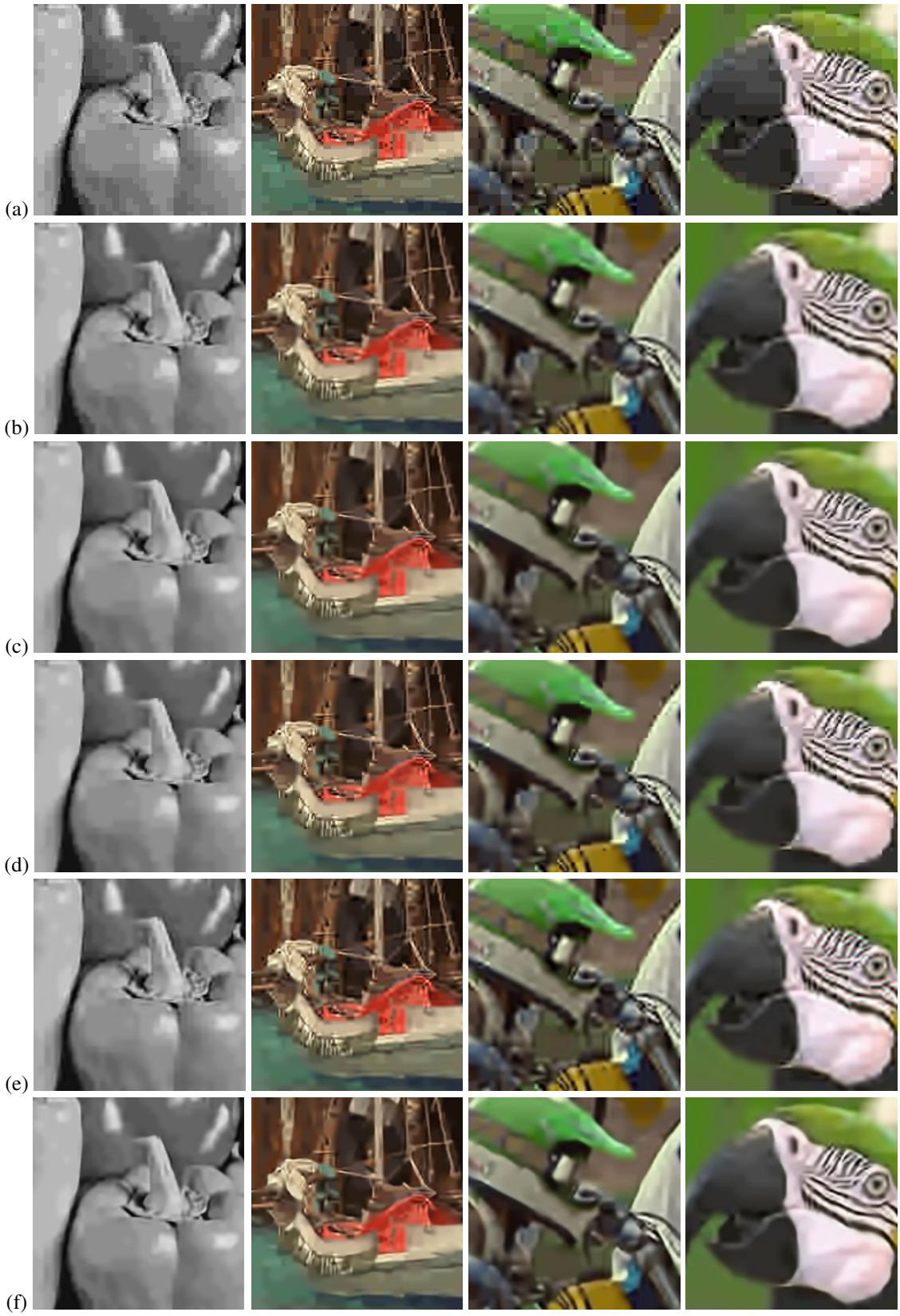

Figure 6: Examples of JPEG artifact suppression (Q3; continued): (a) input JPEG images, (b) re-application of JPEG [10], (c) SADCT [3], (d) FOE [13], (e) PoEdges [5], and (f) our method.

Figure 7: Examples of JPEG artifact suppression on (256×256)-size sub-images (compression quality 7): (a) input JPEG images, (b) SVR-based method [9], (c) SADCT [3], (d) our method, and (e) original un-encoded images.

Figure 8: Examples of JPEG artifact suppression with re-application of JPEG as a preprocessing step (Q2): (a) original JPEG images, (b) input re-application of JPEG [10], (c) output of FOE on input images in (b), and (d) our method.

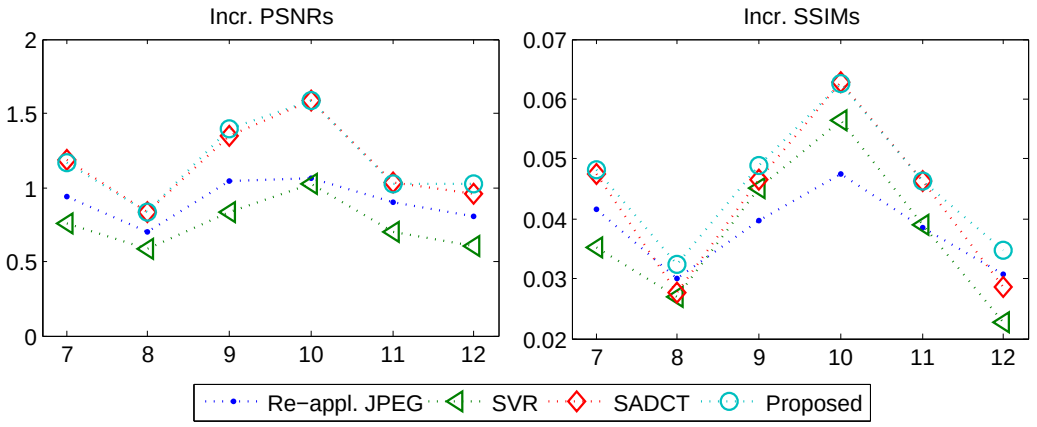

Figure 9: Performance of different JPEG enhancement algorithms for (256×256)-size sub-images (compression quality 7).

Figure 10: Examples of JPEG 2000 artifact suppression: (a) input JPEG 2000 images (0.15BPP), (b) re-application of JPEG [10], (c) PoEdges [5], (d) FOE [13], and (e) our method.

Figure 11: Examples of JPEG 2000 artifact suppression for *Lena* image (0.1BPP): (a) input JPEG 2000 image, (b) Nosratinia [11], (c) FOE [13], (d) PoEdges [5], (e) our method, and (f) original un-encoded image.

Figure 12: Examples of JPEG 2000 artifact suppression (0.1BPP): (a) input JPEG 2000 image, (b) Nosratinia [11], (c) FOE [13], (d) PoEdges [5], (e) our method, and (f) original uncoded images.

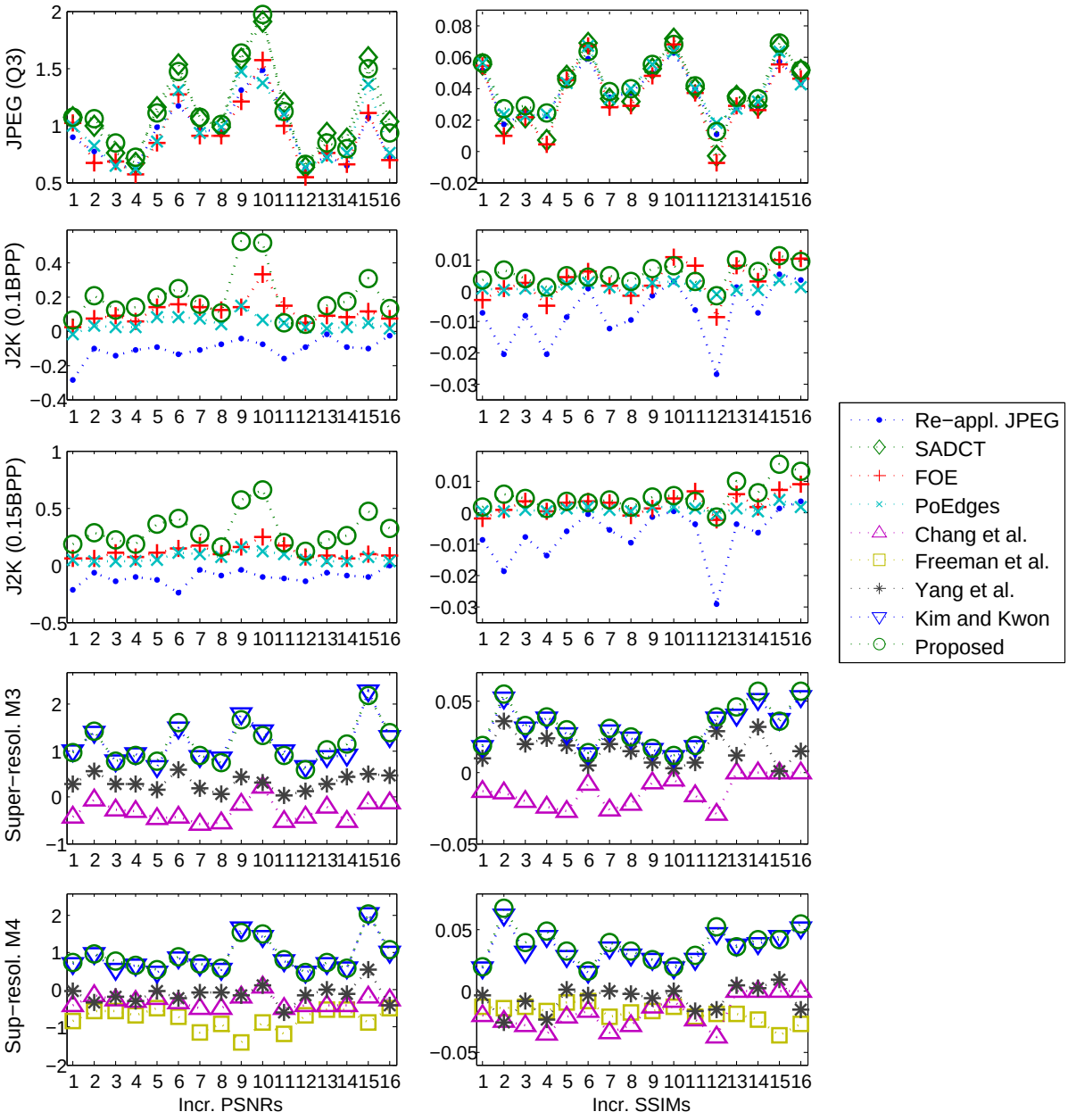

Figure 13: Performance of different image enhancement algorithms on DB1: increases of PSNRs and SSIMs from JPEG images (JPEG; Q3), JPEG 2000 images (JPEG 2000; 0.1BPPs), and bicubic-resampled images (super-resolution). The x axis corresponds to the image index (see Fig. 4 of the main paper).

Figure 14: Examples of image super-resolution for *Lena* image (magnification factor 4): (a) bicubic resampling, (b) Freeman *et al*. [4], (c) He and Siu [6], (d) Kim and Kwon [8], (e) our method, and (f) original high-resolution image.

Figure 15: Examples of image super-resolution (magnification factor 3): (a) bicubic resampling, (b) Chang *et al*. [2], (c) Yang *et al*. [15], (d) Kim and Kwon [8], (e) our method, and (f) original high-resolution images.

Figure 16: Examples of image super-resolution (magnification factor 4): (a) bicubic resampling, (b) Chang *et al*. [2], (c) Freeman *et al*. [4], (d) He and Siu [6], (e) Kim and Kwon [8], (f) our method, and (g) original high-resolution images.

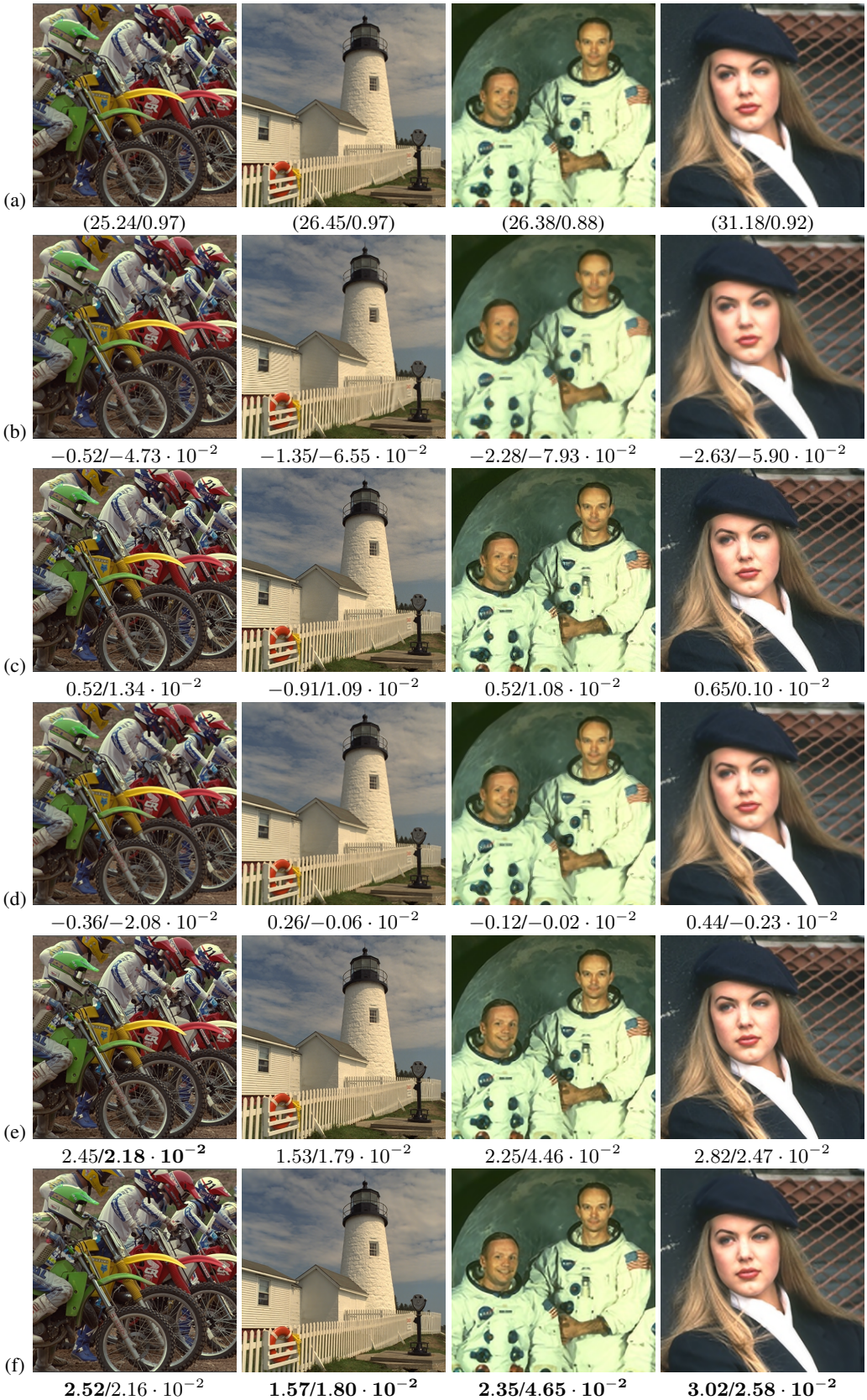

Figure 17: Super-resolution results as per Figure 4 in the main paper, but uncropped — please zoom with the electronic version! (a) Bicubic resampling, (b) Chang *et al*. [2], (c) Freeman *et al*. [4], (d) He and Siu [6], (e) Kim and Kwon [8], and (f) our method. Increases of PSNRs (in dB) and SSIMs with respect to the input bicubic resampled images (displayed below each column) were calculated based on the complete images. For the input images (a), the original PSNR and SSIM values are shown. The best results are marked with bold letters.

Figure 18: JPEG artifact suppression results as per Figure 6 in the main paper, but uncropped — please zoom with the electronic version! (a) Input JPEG images, (b) re-application of JPEG [10], (c) SADCT [3], (d) Edge-perts [5], (e) and (f) FOE [13] applied to (a) and (b), respectively, and (g) our method.

Table 5: Performance of different image enhancement algorithms on DB2: increases of PSNRs (dB) and SSIMs ($\times 10^{-2}$) from JPEG and JPEG2000 images, and bicubic-resampled images (mean and standard deviation). Re-application of JPEG [10], FOE [13], PoEdges [5], SADCT [3], Yang *et al*. [15], Chang *et al*. [2], Freeman *et al*. [4], He and Siu [6], and Kim and Kwon [8]. The best results are marked in bold. For DB2, the mean SSIM improvement of our algorithm is slightly lower than [8].

		[10]	[13]	[5]	[3]	Ours
JPEG (Q3)	PSNR	$0.86(0.28)$	$0.86(0.33)$	$0.90(0.34)$	$\mathbf{1.04(0.37)}$	$1.01(0.37)$
	SSIM	$2.92(1.45)$	$2.59(1.85)$	$3.36(1.41)$	$2.68(1.88)$	$\mathbf{3.23(1.52)}$

			[10]	[13]	[5]	Ours
JPEG2000	0.1BPP	PSNR	$-0.14(0.05)$	$0.10(0.07)$	$0.05(0.03)$	$\mathbf{0.19(0.15)}$
		SSIM	$-0.90(0.92)$	$0.25(0.57)$	$0.11(0.11)$	$\mathbf{0.33(0.29)}$
	0.15BPP	PSNR	$-0.14(0.04)$	$0.10(0.06)$	$0.05(0.04)$	$\mathbf{0.26(0.19)}$
		SSIM	$-0.79(0.84)$	$0.24(0.31)$	$0.09(0.07)$	$\mathbf{0.28(0.29)}$

			[15]	[2]	[4]	[6]	[8]	Ours
Super-resol.	M3	PSNR	$0.24(0.24)$	$-0.46(0.31)$	n/a	n/a	$\mathbf{1.11(0.49)}$	$\mathbf{1.11(0.52)}$
		SSIM	$1.40(0.90)$	$-2.09(1.02)$	n/a	n/a	$2.43(0.96)$	$\mathbf{2.59(1.04)}$
	M4	PSNR	n/a	$-0.42(0.28)$	$-0.51(0.33)$	$-0.14(0.18)$	$0.89(0.43)$	$\mathbf{0.94(0.47)}$
		SSIM	n/a	$-2.77(1.14)$	$-0.09(0.59)$	$-0.58(0.82)$	$3.10(1.07)$	$\mathbf{3.48(1.26)}$

References

[1] P. Breuer, K. I. Kim, W. Kienzle, B. Schölkopf, and V. Blanz. Automatic 3D face reconstruction from single images or video. In *Proc. International Conference on Automatic Face and Gesture Recognition*, pages 1–8, 2008.

[2] H. Chang, D.-Y. Yeung, and Y. Xiong. Super-resolution through neighbor embedding. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 275–282, 2004.

[3] A. Foi, V. Katkovnik, and K. Egiazarian. Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images. *IEEE Trans. Image Processing*, 16(5):1395–1411, 2007.

[4] W. T. Freeman, T. R. Jones, and E. C. Pasztor. Example-based super-resolution. *IEEE Computer Graphics and Applications*, 22(2):56–65, 2002.

[5] P. V. Gehler and M. Welling. Product of "edge-perts". In *Advances in Neural Information Processing Systems*, 2005.

[6] H. He and W.-C. Siu. Single image super-resolution using Gaussian process regression. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 449–456, 2011.

[7] W. Kienzle, G. Bakir, M. Franz, and B. Schölkopf. Face detection - efficient and rank deficient. In *Advances in Neural Information Processing Systems*, pages 673–680, Cambridge, MA, 2005. MIT Press.

[8] K. I. Kim and Y. Kwon. Single-image super-resolution using sparse regression and natural image prior. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32(6):1127–1133, 2010.

[9] V. Laparra, J. Gutiérrez, G. Camps-Valls, and J. Malo. Image denoising with kernels based on natural image relations. *Journal of Machine Learning Research*, 11:873–903, 2010.

[10] A. Nosratinia. Denoising of JPEG images by re-application of JPEG. *Journal of VLSI Signal Processing*, 27(1):69–79, 2001.

[11] A. Nosratinia. Postprocessing of JPEG-2000 images to remove compression artifacts. *IEEE Signal Processing Letters*, 10(10):296–299, 2003.

[12] P. J. Phillips, H. Moon, P. J. Rauss, and S. Rizvi. The FERET evaluation methodology for face recognition algorithms. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000.

[13] S. Roth and M. J. Black. Fields of experts. *International Journal of Computer Vision*, 82(2):205–229, 2009.

[14] T. Wang and G. Zhai. JPEG2000 image postprocessing with novel trilateral deringing filter. *Optical Engineering*, 47(2):027005–1–027005–6, 2008.

[15] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE Trans. Image Processing*, 19(11):2861–2873, 2010.

[16] G. Zhai, W. Lin, J. Cai, X. Yang, and W. Zhang. Efficient quadtree based block-shift filtering for deblocking and deringing. *J. Vis. Commun. Image R.*, 20(8):595–607, 2009.