A Stochastic Image Denoising Algorithm Using 3-D Block Filtering under a Non-local Means Framework

Sreekanth G. Pai and Jiji C. V.
College of Engineering, Trivandrum.

Eighth Indian Conference on Vision, Graphics and Image Processing, ICVGIP 2012

Overview

1. Introduction

Overview

Overview

Overview

Overview

Overview

Image Denoising Problem

$$Y = Y_o + N_\sigma \qquad (1)$$

where Y is a noisy image, Y_o is the original image and N_σ denotes the AWGN noise with variance σ^2

- Reconstruct the original image from its noisy counterpart.
- Preserve the edges, textures and fine details with-in an image.

Image Denoising Algorithms - State-of-the-art

- Denoising based on Soft-thresholding - Donoho [1] in 1995
- Bilateral Filtering - Tomasi *et al*. [2] in 1998
- Non-local Means - Buades *et al*. [3] in 2005
- Block Matching 3-D - Dabov *et al*. [4] in 2007
- Kernel Regression - Takeda *et al*. [5] in 2007
- Stochastic Image Denoising - Estrada *et al*. [6] in 2009
- Patch-based Wiener filtering - Chatterjee *et al*. [7] in 2012

Proposed Stochastic Block Matching

- Integrates 3 different image denoising approaches namely, SID [6],NLM [3] and B3D [4].

Our Contribution

- Introduces random block sampling.
- Block reconstruction using Non-local means framework.
- Residual noise estimation and Iterations.

Steps associated with the algorithm

- Random Sampling and Determination of identical patches
- Filtering in 3-D transform domain
- Weight Assignment and Block Reconstruction
- Residual Noise Estimation and Iteration

Random Sampling and Determination of identical patches

- Segment the noisy image Y into overlapping blocks of fixed size NxN.
- Randomly sample the neigbourhood of each reference block Y_{x_R}.
- Based on a similarity measure identify the most similar blocks.
- Form a 3-D array based on these similar blocks.

$$d(Y_{x_R}, Y_{x_p}) = \frac{\| Y_{x_R}^{th} - Y_{x_p}^{th} \|_2^2}{N^2} \tag{2}$$

$$Y_{x_R}^{th} = \Upsilon(T_{2D}(Y_{x_R}), \lambda_{th2D}) \tag{3}$$

- Υ denotes the hard-thresholding operator.
- T_{2D} denotes the 2-D transform.
- λ_{th2D} denotes the threshold.

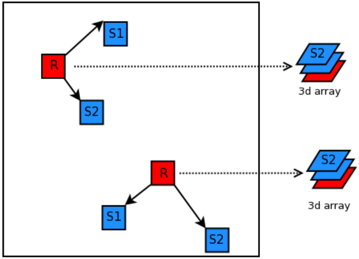

Figure: Random block sampling and forming 3-D array

The 3-D array is denoted by $S_{Y_{x_R}}$

Filtering in 3-D transform domain

- Apply hard-thresholding on the transform domain coefficients.
- Compute the inverse transform.

Denoising of the 3-D array

$$\hat{S}_{Y_{x_R}} = T_{3D}^{-1}(\Upsilon(T_{3D}(S_{Y_{x_R}}), \lambda_{th3D})) \quad (4)$$

- $\hat{S}_{Y_{x_R}}$ is the estimate of $S_{Y_{x_R}}$.
- T_{3D} indicates the 3-D transforms such as 3-D DCT.
- λ_{th3D} indicates the threshold.

Weight Assignment and Block Reconstruction

- Reconstruct the reference block from the 2-D estimates obtained from $\hat{S}_{Y_{x_R}}$ by assigning proper weights.
- Compute the basic estimate by weighted averaging of all the overlapping block-wise estimates.

Weights are assigned based on the equation (5)

$$\omega_{x_R}^{x_p} = \exp\left\{-\frac{\|Y_{x_p} - Y_{x_R}\|_2^2}{h^2}\right\} \tag{5}$$

Denoised Basic Estimate

$$\hat{Y}(x) = \frac{\sum_{Y_{x_R}} \sum_{Y_{x_p}} \omega_{x_R}^{x_p} \hat{S}_{Y_{x_R}}}{\sum_{Y_{x_R}} \sum_{Y_{x_p}} \omega_{x_R}^{x_p}} \tag{6}$$

Residual Noise Estimation and Iteration

A robust median estimator [8] is used to estimate the noise level at the output.

$$\sigma = \frac{Median(|H_{ij}|)}{0.6745} \tag{7}$$

- H_{ij} represents the subband HH_1 of the noisy image.
- The estimated σ and the basic estimates are used as inputs to the algorithm and the steps are repeated.

Output of SBM

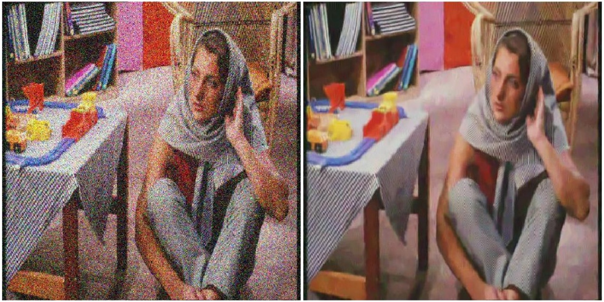

Figure: Noisy($\sigma = 35$) Barbara Image and the corresponding estimate obtained for Stochastic Block matching algorithm

Denoising performance for various values of sigma

Figure: Denoising performance of noisy lena image for various noise levels (a) Noisy lena input ($\sigma = 20$) (b) $\sigma = 30$ (c) $\sigma = 40$ (d) $\sigma = 50$ (e), (f), (g), (h) are the corresponding SBM Outputs obtained after 3 iterations

Denoising performance for various values of sigma

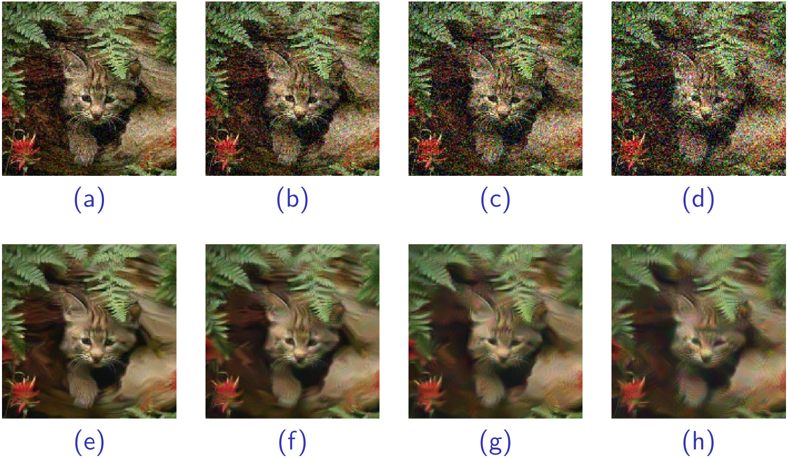

Figure: Denoising performance of noisy cat image for various noise levels (a) Noisy cat input ($\sigma = 20$) (b) $\sigma = 30$ (c) $\sigma = 40$ (d) $\sigma = 50$ (e), (f), (g), (h) are the corresponding SBM Outputs obtained after 3 iterations

Comparison with state-of-the-art algorithms

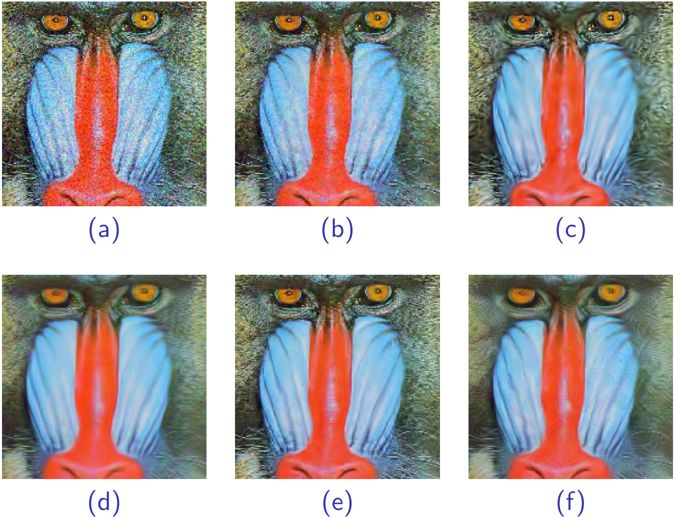

Figure: Comparison of denoising results on noisy mandrill image corrupted by AWGN of $\sigma = 25$ (a) Noisy input (b) SID [6] (c) SKR [5] (d) PLOW [7] (e) B3D [4] (f) SBM

Comparison with state-of-the-art algorithms

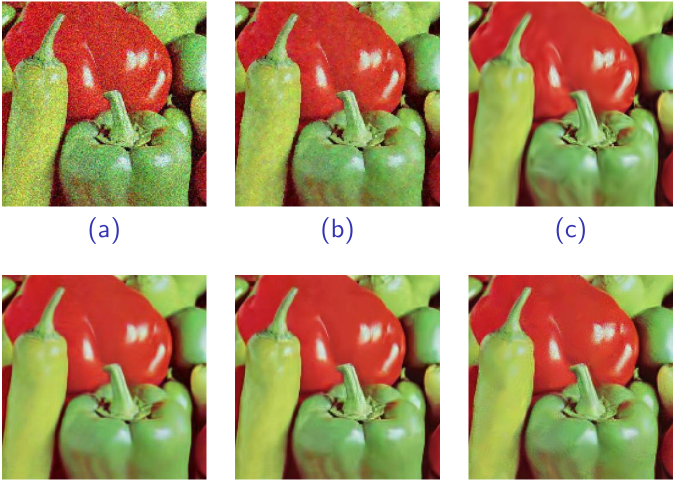

Figure: Comparison of denoising results on noisy peppers image corrupted by AWGN of $\sigma = 25$ (a) Noisy input (b) SID [6] (c) SKR [5] (d) PLOW [7] (e) B3D [4] (f) SBM

Table: PSNR Comparisons

Image	σ	BF	SID	SKR	PLOW	B3D	SBM
Lena	10	23.6260	29.8356	28.0067	31.9649	33.6566	**32.7866**
256 x 256	15	21.0820	27.0729	28.3022	30.5604	32.1554	**30.8783**
	25	17.3951	23.5551	28.4248	28.5314	30.1508	**28.3701**
Mandrill	10	23.9305	28.6300	22.4450	24.7814	29.9350	**28.8217**
256 x 256	15	20.1069	25.4115	22.6243	23.4859	27.4718	**25.6156**
	25	16.4750	20.9049	22.6392	22.0526	24.6999	**22.9131**
Peppers	10	30.8810	29.7871	26.7737	31.6541	32.7431	**32.0083**
256 x 256	15	25.8521	27.9560	27.4036	30.1817	31.2136	**29.8878**
	25	19.8231	23.6217	27.8698	28.1219	29.0092	**27.0007**
BSD	10	29.7752	29.5463	26.4384	29.5447	33.9796	**32.7523**
Image	15	27.2750	26.9026	26.5559	28.1701	31.6922	**28.7059**
	25	23.7925	23.5109	26.0232	25.9148	28.6905	**24.8406**

Table: Structural Similarity Index Comparisons

Image	σ	BF	SID	SKR	PLOW	B3D	SBM	Noisy
Lena 256 x 256	10	0.8783	0.8841	0.8165	0.9021	0.9394	**0.9175**	0.8199
	15	0.8420	0.7847	0.8247	0.8840	0.9130	**0.8872**	0.7001
	25	0.7251	0.7051	0.8323	0.8440	0.8780	**0.8396**	0.5222
Mandrill 256 x 256	10	0.9299	0.9528	0.6208	0.7522	0.9493	**0.9361**	0.9327
	15	0.8648	0.8833	0.6463	0.6990	0.9088	**0.8719**	0.8688
	25	0.7345	0.7539	0.6899	0.6159	0.8282	**0.8072**	0.7405
Peppers 256 x 256	10	0.8992	0.8869	0.8220	0.8998	0.9245	**0.9013**	0.8292
	15	0.8563	0.7573	0.8293	0.8838	0.9001	**0.8741**	0.7085
	25	0.7122	0.7434	0.8285	0.8479	0.8664	**0.8321**	0.5296
BSD Image	10	0.8981	0.9203	0.7802	0.9128	0.9641	**0.9391**	0.8992
	15	0.8575	0.8667	0.7889	0.8855	0.9392	**0.9138**	0.8185
	25	0.7737	0.7655	0.7962	0.8155	0.8831	**0.8305**	0.6710

Table: Output PSNR and estimated σ obtained for three iterations for a lena image

σ	PSNR			Estimated σ		
	SBM_1	SBM_2	SBM_3	SBM_1	SBM_2	SBM_3
10	32.5396	32.7884	32.7866	3.8948	2.3035	1.7801
15	30.3511	30.8307	30.8783	4.6694	2.4466	1.8414
20	28.7579	29.3962	29.4758	5.4239	2.9207	2.3180
25	27.4587	28.2797	28.3701	6.4466	3.1520	2.3804
50	23.2955	24.7667	24.8850	10.9295	4.1920	2.9791

Table: Noise deviation estimated from the output of different algorithms

Algorithm	σ	Lena	Peppers	F16	Baboon	Cat
SKR	10	0.2848	0.2876	0.2812	0.4961	0.3898
	15	0.2869	0.2959	0.3090	0.5640	0.4291
	20	0.3144	0.3243	0.3431	0.6842	0.4724
	25	0.3365	0.3541	0.4012	0.8716	0.5903
PLOW	10	0.4535	0.4739	0.4590	1.0768	1.1271
	15	0.4802	0.3687	0.1550	0.9516	1.0802
	20	0.5938	0.5415	0.2759	0.8563	1.0457
	25	0.8458	0.9662	0.7165	0.8687	1.0361
B3D	10	0.9610	1.3362	0.8627	9.8456	7.3105
	15	0.8053	0.9485	0.8223	7.9590	5.8473
	20	0.8117	0.8526	0.7852	5.6121	4.0021
	25	0.7072	0.7772	0.7862	4.3690	2.9314
SBM	10	1.4121	1.2318	1.3841	1.4181	1.3108
	15	1.2416	1.5183	1.6181	1.4617	1.5782
	20	1.5791	1.3892	1.4295	1.6729	1.5191
	25	1.3186	1.4481	1.6721	1.5684	1.5311

Histogram of the result obtained by subtracting the denoised estimate from original image

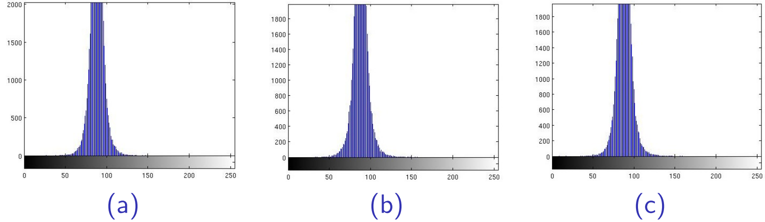

Figure: Histogram of the result obtained by subtracting the denoised estimate from original image (a) Histogram of R Channel (b) Histogram of G Channel (c) Histogram of B Channel

Conclusion

- Stochastic Block Matching (SBM) integrates non-local means with random sampling and block matching.
- SBM outperforms most of the other state-of-the-art algorithms such as SKR [5], PLOW [7], SID [6].
- SBM is comparable with B3D algorithm.
- Scope for improvement in terms of PSNR is more for SBM compared to other state-of-the-art.

[1] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Trans. Inform. Theory*, vol. 41, pp. 613–627, May 1995.

[2] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," *in Proc. IEEE Int. Conf. Comp. Vis.*, pp. 839–846, 1998.

[3] A. Buades, B. Coll, and J. Morel, "A non-local algorithm for image denoising," *in Proc. IEEE Int. Conf. Comp. Vis. Pat. Recog.*, vol. 2, pp. 60–65, 2005.

[4] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, pp. 2080–2095, Aug. 2007.

[5] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Trans. Image Processing*, vol. 16, pp. 349–366, Feb. 2007.

[6] F. Estrada, D. Fleet, and A. Jepson, "Stochastic image denoising," *in Proc. BMVC*, Sept. 2009.

[7] P. Chatterjee and P. Milanfar, "Patch-based near-optimal image denoising," *IEEE Trans. Image Processing*, vol. 21, pp. 1635–1649, April 2012.

[8] S. G. Chang, B. Yu, and M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression," *IEEE Trans. Image Processing*, vol. 9, pp. 1532–1546, Sept. 2000.