Optimal Binary Descriptor for Pre-Classification in Efficient Non-Local Mean Image Denoising

Jan-Ray Liao^{a,b,*}, Yi-An Chen^a

^aGraduate Institute of Communication Engineering, National Chung Hsing University, Taichung, Taiwan

^bDepartment of Electrical Engineering, National Chung Hsing University, Taichung, Taiwan

*Corresponding Author: jrliao@mail.nchu.edu.tw

Abstract

With the popularity of consumer digital imaging devices, image denoising has become an important issue in image processing. Among currently available image denoising algorithms, non-local mean (NLM) is one of the most effective methods. NLM calculates weights of neighboring pixels based on the similarity between two image patches. The pixel is then estimated by the weighted sum of the neighboring pixels. Because the large number of the patches needs to be compared, NLM incurs high computational cost and this makes it impractical in many situations. On the other hand, a binary descriptor produces a simple binary string to describe an image patch by comparing pixels with a given threshold. Comparing binary descriptors between two image patches is much more efficient than comparing the two patches directly. Preliminary results show that using binary descriptors to reject dissimilar patches from the computation can significantly improve the performance of NLM. In this paper, we seek to find the best binary descriptor which can provide large speedup and enhance denoising performance. Experimental results show that using two types of 18-neighbor binary descriptors can effectively increase the denoising performance of NLM and significantly reduce the execution time.

Keywords: image denoising, non-local mean, binary descriptor, pre-classification.

1. Introduction

With the popularity of consumer digital imaging devices, image denoising has become an important application in image processing especially when images are

acquired in undesirable condition such as in poor illumination or under high ISO setting. One of the most effective image denoising methods currently available is the non-local mean (NLM)^(1,2) method. For a pixel to be denoised, NLM first calculates weights for its neighboring pixels based on the similarity between the image patch of the current pixel and the image patches of the neighboring pixels. The current pixel is then replaced by the weighted sum of its neighboring pixels and itself. Because pixels from similar patches are given large weights, the collection of similar pixels can effectively remove unwanted noise if a large number of image patches are considered.

However, the area of the neighborhood to be searched needs to be sufficiently large so that sufficient number of image patches are evaluated and enough similar patches can be collected. This means that NLM incurs extremely high computational cost and there are many researches in speeding up the computation. These acceleration methods can be divided into three categories. The first category is to reject dissimilar patches by comparing simplified parameters such as: means⁽³⁻⁵⁾, variances^(4,5), higher-order statistical moments^(6,7), gradients⁽³⁾, and pixel intensities^(8,9). The second category is to simplify patch comparison by reducing the dimensions of the image patches⁽¹⁰⁻¹⁵⁾ by using principle component analysis^(10,11,13-15) and singular value decomposition⁽¹²⁾. The third category is to accelerate the computation through either efficient algorithms or data structures⁽¹⁶⁻²³⁾. The efficient algorithms being used are random selections^(16,23), fast Fourier transform^(17,18,22), squared image^(18,19), and probabilistic early termination⁽²¹⁾. The data structures for improving efficiency include Laplacian pyramid⁽¹⁸⁾ and multiresolution representations⁽²⁰⁾.

Among these three categories of acceleration techniques, the first category has an additional advantage of

producing images with less blurring. This is because that the small weights given to pixels from dissimilar patches eventually accumulate to a significant amount and bias the estimation⁽²⁴⁾. Rejecting dissimilar patches can remove these small weights from computation and produce less blurred image. However, additional statistical parameters such as means or gradients need to be calculated before the NLM process and this adds additional complexity in the algorithm. In a preliminary study⁽²⁵⁾, we have shown that using binary descriptors in the pre-classification can significantly improve NLM in both reducing computation time and enhancing denoising performance. A binary descriptor of an image patch compares the pixels in the patch with a given threshold and produces a simple binary string to describe the patch. The advantages of binary descriptors are that they are very simple to generate and the cost of comparison between two descriptors is extremely low. Therefore, the computation overhead is very low.

Encouraged by these preliminary results, we seek to find the best binary descriptor for NLM in this paper by extensively testing various binary descriptors. Overall, 50 binary descriptors have been tested. Experimental results show that using two types of 18-neighbor binary descriptors can effectively increase the denoising performance of NLM and significantly reduce the execution time. Therefore, our conclusion is that these two types of binary descriptors are the optimal choice in NLM pre-classification.

The rest of the paper is organized as follows. In the second section, we give a brief introduction on NLM algorithm and present the tested binary descriptors. In the third section, we show the experimental results. In the fourth section, we conclude.

2. Non-Local Mean and Binary Descriptors

2.1 Non-Local Mean

Let X be the original noise-free image, $Y = X + N$ be a noisy image with an additive Gaussian noise N of zero mean and variance σ^2 , and $\hat{X}$ be the denoised image estimated from Y . We assume that the pixel values of all the images are normalized to the range between 0 and 1. Let $\mathbf{m} = (x, y)$ and $\mathbf{n} = (x', y')$ denote two locations in the images. Let $Y(\mathbf{m})$ denote the pixel value at location $\mathbf{m}$ and $\mathbf{Y}_{\mathbf{n}}$ denote the image patch which is centered at location $\mathbf{n}$. The patch $\mathbf{Y}_{\mathbf{n}}$ extends to k pixels around $\mathbf{n}$ in all four directions (top, bottom, left, and right), i.e., the

size of the patch is $(2k+1)^2$.

For a location $\mathbf{m}$, let $S_{\mathbf{m}}$ be a square search window centered at $\mathbf{m}$ and extended to s pixels in all four directions, i.e., the size of the search window is $(2s+1)^2$. For each patch $\mathbf{Y}_{\mathbf{n}}$ where $\mathbf{n} \in S_{\mathbf{m}}$, the weight is calculated as:

$$w_{\mathbf{nm}} = \exp\left(-\frac{\|\mathbf{Y}_{\mathbf{n}} - \mathbf{Y}_{\mathbf{m}}\|^2}{h^2}\right) \quad (1)$$

where $\|\bullet\|^2$ is the Euclidean norm and h is an adjustable parameter.

The estimated pixel value is the weighted mean:

$$\hat{X}(\mathbf{m}) = \frac{\sum_{\mathbf{n} \in S_{\mathbf{m}}} w_{\mathbf{nm}} Y(\mathbf{n})}{\sum_{\mathbf{n} \in S_{\mathbf{m}}} w_{\mathbf{nm}}} \quad (2)$$

2.2 Binary Descriptors

A binary descriptor of an image patch compares the pixels within the patch with a given threshold and produces an integer to describe the patch.

Let $q(\bullet)$ be a binary operator (i.e., a quantizer) defined as below:

$$q(Y(\mathbf{n})) = \begin{cases} 1, & Y(\mathbf{n}) \geq T_h \\ 0, & Y(\mathbf{n}) < T_h \end{cases} \quad (3)$$

Based on previous experimental results⁽²⁵⁾, T_h is set to 0.5 and we called this descriptor “simple binary pattern” (SBP).

The relative positions of the pixels to be considered in the description with respect to the center of the patch are given by a set of vectors V :

$$V = \{\mathbf{v}_i = (dx_i, dy_i), i = 1, \dots, K\}, \quad (4)$$

For example, if the center of patch and its four neighbors are considered in the description, the set V is:

$$V = \{(0,0), (-1,0), (1,0), (0,-1), (0,1)\}. \quad (5)$$

If N neighbors are considered, we called the descriptor a “ N -neighbor description”. Therefore, the above example is a 4-neighbor description.

The description for a patch $\mathbf{Y}_{\mathbf{m}}$ is calculated as:

$$D(\mathbf{Y}_m) = \sum_{i=1}^K q(Y(\mathbf{m} + \mathbf{v}_i)). \quad (4)$$

To put it simply, the description calculates the number of ones within the pixels being considered and it can be as a quantized mean estimator.

For our NLM algorithm, only when $D(\mathbf{Y}_m)$ is equal to $D(\mathbf{Y}_n)$, the weight w_{nm} in (1) is calculated. Otherwise, the weight w_{nm} is set to zero. In other words, only patches with the same binary description are included in the denoising calculation of (2).

2.3 Tested Binary Descriptors

Although preliminary study⁽²⁵⁾ shows that 12-neighbor and 16-neighbor SBP's produce better results than other descriptors, there are only a very limited number of descriptors being tested in the study. In order to comprehensively understand the behavior of SBP, we tested five kinds of SBP based on the number of neighbors being included in the description. They are: 10-neighbor, 12-neighbor, 14-neighbor, 16-neighbor, and 18-neighbor SBP's. For each kind of SBP, 10 variations on the relative positions of neighboring pixels are studied. Overall, 50 SBP's have been tested. Fig. 1~5 shows the representative variations which have better performance than the other SBP's being tested.

Fig. 1. The relative positions of the neighboring pixels consider in the 10-neighbor SBP: (a) Type-A, (b) Type-B, (c) Type-C, and (d) Type-D.

Fig. 2. The relative positions of the neighboring pixels consider in the 12-neighbor SBP: (a) Type-A, (b) Type-B, (c) Type-C, (d) Type-D, (e) Type-E, and (f) Type-F.

Fig. 3. The relative positions of the neighboring pixels consider in the 14-neighbor SBP: (a) Type-A, (b) Type-B, (c) Type-C, and (d) Type-D.

Fig. 4. The relative positions of the neighboring pixels consider in the 16-neighbor SBP: (a) Type-A, (b) Type-B, (c) Type-C, (d) Type-D, (e) Type-E, (f) Type-F, (g) Type-G, (h) Type-H, and (i) Type-I.

3. Experimental Results

The SBP pre-classified NLM algorithm was compared with the original NLM algorithm^(1,2) and the accelerated NLM algorithm using the mean of the image patch for

classification⁽³⁻⁵⁾. For all algorithms, h is set to $100\sigma^2$. The mean of a patch is calculated over a 7×7 neighborhood and weights are only calculated for patches with means within $\pm 10\%$ of the mean of the current patch. The reason that mean-classified NLM is chosen for comparison is that the SBP can be viewed as an approximate estimator of the pixel sum and it is directly proportional to the mean.

Images from USC-SIPI image database were tested under the platform of a personal computer with 3.6 GHz Intel Core i7-4790 CPU and 8 GB of memory. All algorithms are implemented in C under Microsoft Visual Studio Community 2015.

For the 10-neighbor SBP in Fig. 1, type-10B and type-10C are better and the performances of both descriptors are almost the same because they are symmetric. For the 12-neighbor SBP in Fig. 2, type-12D is better. For the 14-neighbor SBP in Fig. 3, type-14B and type-14D are better. When the number of neighbors in the descriptor increases, there are more variations in the performance among different types. For the 16-neighbor SBP in Fig. 4, type-16F and type-16H are better for most cases but type-16A, type-16B, type-16E, and type-16G are better for a few cases. For the 18-neighbor SBP in Fig. 5, type-18C and type-18D are better for most cases but type-18A and type-18B are better for some cases.

Fig. 6 shows PSNR vs. the standard deviation σ of the additive Gaussian noise for four of the images tested. Since the differences among the better-performed SBP's with the same number of neighbors are small, we only show the

Fig. 5. The relative positions of the neighboring pixels consider in the 18-neighbor SBP: (a) Type-A, (b) Type-B, (c) Type-C, and (d) Type-D.

Fig. 6. PSNR vs. standard deviation of additive Gaussian noise for four images: (a) Baboon, (b) F-16, (c) House, and (d) Lena.

results of one representative SBP for the same number of neighbors. From Fig. 6, we can see SBP-classified NLM generally performs better than the original NLM except for the image Lena. Of all the SBP's tested, type-18C and type-18D SBP's are the best for most situations. The largest gap between type-18D SBP NLM and the original NLM is 2.3949 dB which occurs at $\sigma=20$ for the image Baboon. In the worst case, PSNR of type-18D SBP NLM is 0.4809 dB lower than that of the original NLM which happens at $\sigma=60$ for the image Lena. Comparing with the mean-classified NLM, our SBP-classified NLM is better at small σ but worse at large σ . This could be attributed to the fact that the description is less accurate when noise is large. The best gap between type-18D SBP NLM and the mean-classified NLM is 1.5008 dB at $\sigma=20$ for the image Baboon and the worst gap is -2.8293 dB at $\sigma=60$ for the image Lena. However, from Fig. 7 where the average execution time is shown, we can see that the execution time of the type-18D SBP NLM is 38.36% of the mean-classified NLM and 23.81% of the

original NLM. Therefore, SBP-based NLM is much more computationally efficient than the other algorithms.

Fig. 8 shows PSNR vs. the number of neighbors in the descriptors for $\sigma=20$ and $\sigma=60$. From the figure, we can see that the differences among different descriptors are small when noise is small. However, when noise is large ($\sigma=60$), performance becomes better as the number of neighbors increases but the improvement flattens out as the number of neighbors reaches beyond 16. The largest improvement is 0.7502 dB from type-10B to type-18D for the image Lena.

Fig. 9 shows the images produced by the original NLM, the mean-classified NLM, the type-10B SBP NLM, and the type-18D NLM when $\sigma=40$. From the figures, we can see that SBP-classified NLM can preserve image details better and it is still competitive for the image Lena despite the fact that its PSNR is not better than the other two algorithms.

From these results, we can observe that neighbors whose distance are closer to the center of the patch are more important to the denoising performance than neighbors that

Fig. 7. The average execution time for all the methods

are further away from the center. As the number of neighbors increases, there are more neighbors of the same distance to choose from and the choices of the best outlying neighbors depend more on the characteristics of the image. Summarizing from the experimental results, our conclusion is that type-18C and type-18D SBP's are the optimal choices.

4. Conclusions

In this paper, the optimal choice of using simple binary descriptor to reject dissimilar patches in NLM has been studied through extensive experiments. Experimental results show that using two types of 18-neighbor simple binary descriptors can effectively maintain the denoising performance of NLM while significantly reduce the execution time. Therefore, our conclusion is that these two types of binary descriptors are the optimal choice in NLM pre-classification.

Acknowledgment

This work was supported by Ministry of Science and Technology, Taiwan, Republic of China, under contract MOST 104-2221-E-005-052.

References

- (1) A. Buades, B. Coll, and J. M. Morel, "A Review of Image Denoising Methods, with a New One," Multiscale Modeling and Simulation, Vol. 4, No. 2, pp. 490-530, 2005.
- (2) A. Buades, B. Coll, and J. M. Morel, "A Non-Local Algorithm for Image Denoising," in Proc. IEEE Computer Vision and Pattern Recognition (CVPR), Vol. 2, Vol. 4191, pp. 60-65, 2005.
- (3) M. Mahmoudi and G. Sapiro, "Fast Image and Video Denoising via Nonlocal Means of Similar Neighborhoods," IEEE Signal Processing Letters, Vol. 12, No. 12, pp. 839-842, 2005.
- (4) C. Kervrann, J. Boulanger, and P. Coupe, "Bayesian Non-Local Means Filter, Image Redundancy and Adaptive Dictionaries for Noise Removal," Proc. 1st Int. Conf. Scale Space and Variational Methods in Computer Vision (SSVM), Vol. 4485, pp. 520-532, 2007.
- (5) T. Brox, O. Kleinschmidt, and D. Cremers, "Efficient Nonlocal Means for Denoising of Textural Patterns," IEEE Trans. Image Processing, Vol. 17, No. 7, pp. 1083-1092, 2008.
- (6) P. Coupe, P. Yger, and C. Barillot, "Fast Non Local Means Denoising for 3D MR Images," in Proc. Medical Image Computing and Computer-Assisted Intervention (MICCAI), Vol. 4191, pp. 33-40, 2006.

(a)

(b)

Fig. 8. PSNR vs. number of neighbors in the descriptor: (a) $\sigma = 20$ and (b) $\sigma = 60$.

(a)

(b)

(c)

(d)

Fig. 4. Images generated by different NLM algorithms. From left to right, the methods used are the original NLM, the mean-classified NLM, type-10B SBP NLM, and type-18D SBP NLM: (a) Baboon, (b) F-16, (c) House, and (d) Lena.

- (7) A. Dauwe, B. Goossens, H. Luong, and W. Philips, "A Fast Non-Local Image Denoising Algorithm," Proc. SPIE, Vol. 6812, pp. 1331-1334, 2008.
- (8) R. C. Bilcu and M. Vehvilainen, "Combined Non-Local Averaging and Intersection of Confidence Intervals for Image De-noising," Proc. IEEE Int. Conf. Image Processing, pp. 1736-1739, 2008.
- (9) C. Pang, O. C. Au, J. Dai, Y. Wen, and Z. Feng, "A Fast NL-Means Method in Image Denoising Based on the Similarity of Spatially Sampled Pixels," Proc. IEEE Int.

- Workshop Multimedia Signal Processing, pp. 1-4, 2009.
- (10) N. Azzabou, N. Paragios, and F. Guichard, "Image Denoising Based on Adapted Dictionary Computation," in *Proc. IEEE Int. Conf. Image Processessing (ICIP)*, Vol. 3, pp. 109-112, 2007.
 - (11) T. Tasdizen, "Principal Components for Non-Local Means Image Denoising," *Proc. Int. Conf. Image Processing*, pp. 1728-1731, 2008.
 - (12) J. Orchard, M. Ebrahimi, and A. Wong, "Efficient Nonlocal-Means Denoising Using the SVD," *Proc. IEEE Int. Conf. Image Processing*, pp. 1732-1735, 2008.
 - (13) T. Tasdizen, "Principal Neighborhood Dictionaries for Nonlocal Means Image Denoising," *IEEE Trans. Image Processing*, Vol. 18, No. 12, pp. 2649-2660, 2009.
 - (14) T. Thaipanich, B. T. Oh, P.-H. Wu, D. Xu, C.-C. J. Kuo, "Improved Image Denoising with Adaptive Nonlocal Means (ANLmeans) Algorithm," *IEEE Trans. Consumer Electronics*, Vol. 56, No. 4, pp. 2623-2630, 2010.
 - (15) D. Van De Ville and M. Kocher, "Nonlocal Means with Dimensionality Reduction and SURE-Based Parameter Selection," *IEEE Trans. Image Processing*, Vol. 20, No. 9, pp. 2683-2690, 2011.
 - (16) N. Azzabou, N. Paragios, F. Guichard, "Random Walks, Constrained Multiple Hypothesis Testing and Image Enhancement," *Proc. European Conference on Computer Vision (ECCV)*, Vol. 1, pp. 379-390, 2006.
 - (17) J. Wang, Y. Guo, Y. Ying, Y. Liu, and Q. Peng, "Fast Non-Local Algorithm for Image Denoising," *Proc. IEEE Int. Conf. Image Processing*, pp. 1429-1432, 2006.
 - (18) Y. L. Liu, J. Wang, X. Chen, Y. W. Guo, and Q. S. Peng, "A Robust and Fast Non-Local Means Algorithm for Image Denoising," *Journal of Computer Science and Technology*, Vol. 23, No. 2, pp. 270-279, 2008.
 - (19) J. Darbon, A. Cunha, T. Chan, S. Osher, and G. Jensen, "Fast Nonlocal Filtering Applied to Electron Cryomicroscopy," *Proc. 5th IEEE Int. Symp. Biomedical Imaging (ISBI)*, pp. 1331-1334, 2008.
 - (20) V. Karnati, M. Uliyar, and S. Dey, "Fast Non-Local Algorithm for Image Denoising," *Proc. IEEE Int. Conf. Image Processing*, pp. 3873-3876, 2009.
 - (21) R. Vignesh, B. T. Oh, and C.-C. Kuo, "Fast Non-Local Means (NLM) Computation with Probabilistic Early Termination," *IEEE Signal Process. Letters*, Vol. 17, No. 3, pp. 277-280, 2010.
 - (22) C. A. Deledalle, V. Duval, and J. Salmon, "Non-Local Methods with Shape-Adaptive Patches," *Journal of Mathematical Imaging and Vision*, Vol. 43, No. 2, pp. 103-120, 2012.
 - (23) S. H. Chan, T. Zickler, and Y. M. Lu, "Monte Carlo Non Local Means: Random Sampling for Large-Scale Image Filtering," *IEEE Trans. Image Processing*, Vol. 23, No. 8, pp. 3711-3725, 2014.
 - (24) J. Salmon, "On Two Parameters for Denoising with Non-Local Means," *IEEE Signal Processing Letters*, Vol. 17, No. 3, pp. 269-272, 2010.
 - (25) J.-R. Liao, Y.-W. Tsai, Y.-H. Huang, "Efficient Non-Local Image Denoising Using Binary Descriptor Classification," *Proc. 4th IIAE Int. Conf. Industrial Application Engineering*, pp. 303-308, 2016.
 - (26) T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 24, No. 7, pp. 971-987, 2002.