Efficient Non-Local Means Image Denoising Using Binary Descriptor Pre-Classification and Distance-Based Pre-Termination

Jan-Ray Liaoa,b,*, Shing-Yu Yehb

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

Among currently available image denoising algorithms, non-local means (NLM) is one of the most effective methods. NLM calculates weights of neighboring pixels based on the similarity between two image patches. The denoised pixel is estimated by the weighted sum of the neighboring pixels. Because the number of neighboring patches needs to be sufficiently large so that enough similar patches can be covered, NLM incurs high computational cost. In this paper, we incorporate binary descriptor pre-classification and distance-based pre-termination to exclude dissimilar patches from the computation so that the performance of NLM can be enhanced. A binary descriptor produces a simple binary string to describe an image patch by comparing pixels with a given threshold. Before calculating the weight between two patches, their binary descriptors are compared and the patch is skipped if the binary descriptors are not the same. Although binary descriptor pre-classification can exclude a large number of dissimilar patches, some patches with large distance between them are still calculated. Therefore, after the pre-classification, during the process of calculating distance between two patches, the computation is pre-terminated if the accumulated distance is larger than a threshold and that patch is also excluded from the weight calculation. Experimental results show that combining these two approaches in excluding dissimilar patches can effectively increase the denoising performance of NLM and significantly reduce the execution time.

Keywords: image denoising, non-local means, binary descriptor pre-classification, distance-based pre-termination.

1. Introduction

Image denoising is an important step in the process of image acquisition especially under undesirable conditions such as poor illumination. One of the most effective image denoising algorithms currently available is non-local means (NLM)(1,2). For a pixel to be denoised, NLM first calculates weights for each of its neighboring pixels based on the similarity between the image patch of the current pixel and the image patches of the neighboring pixels. The output is the weighted sum of the current pixel and all its neighboring pixels. Because pixels from similar patches are given larger weights, the averaging of similar pixels can effectively remove unwanted noise if sufficient number of similar image patches are covered in the calculation. In order to collect enough similar patches, the area of the searched neighborhood needs to be sufficiently large. This means that a large number of weights need to be calculated for a single pixel. Thus, NLM incurs extremely high computational cost and there are many researches in accelerating the computation.

The methods to accelerate NLM 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) such as using principle component analysis(10,11,13-15) or singular value decomposition(12). The third category is to directly accelerate the computation. This can be accomplished by using efficient algorithms such as...
random selections\(^{16,17}\), fast Fourier transform\(^{18-20}\), squared image\(^{19,21}\), or probabilistic early termination\(^{22}\). Data structures are also used to improve efficiency such as Laplacian pyramid\(^9\) or multiresolution representations\(^{23}\).

Among these three categories, dissimilar patch rejection of the first category has an additional benefit 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\(^{24}\). However, additional statistical parameters such as means or gradients need to be calculated and this computation adds additional complexity in the algorithm.

The motivation of this paper is to use two simple approaches to replace previous methods of rejecting dissimilar patches. The first is binary descriptor pre-classification. A binary descriptor of a 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. Before calculating the weight between two patches, their binary descriptors are compared and the patch is skipped if the binary descriptors are not the same. The second is distance-based pre-termination. Although binary descriptor pre-classification can exclude a large number of dissimilar patches, some patches with large distance between them still pass the pre-classification. Therefore, after the descriptor comparison, during the process of calculating distance between two patches, the distance computation is pre-terminated if the accumulated distance is larger than a threshold and that patch is also excluded from the weight calculation. Experimental results show that combining these two approaches in excluding dissimilar patches can effectively increase the denoising performance of NLM and significantly reduce the execution time.

The rest of the paper is organized as follows. In the second section, we present the NLM algorithm, the binary descriptor, and the proposed NLM algorithm. In the third section, we show the experimental results. In the fourth section, we conclude.

## 2. Non-Local Means and Binary Descriptors

### 2.1 Non-Local Means

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 $\sigma^2$, 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 $m = (x, y)$ and $n = (x', y')$ denote two locations in the images. Let $Y(m)$ denote the pixel value at location $m$ and $Y_n$ denote the image patch which is centered at location $n$. The patch $Y_n$ extends to $k$ pixels around $n$ in all four directions (top, bottom, left, and right), i.e., the size of the patch is $(2k+1)^2$.

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

$$w_{nm} = \exp \left( - \frac{||Y_n - Y_m||_2^2}{h^2} \right)$$  \hspace{1cm} (1)

where $||\cdot||_2^2$ is the Euclidean norm and $h$ is an adjustable parameter.

The estimated pixel value is the weighted mean:

$$\hat{X}(m) = \frac{\sum_{n \in S_m} w_{nm} Y(n)}{\sum_{n \in S_m} w_{nm}}.$$  \hspace{1cm} (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 $q(\cdot)$ be a binary operator (i.e., a quantizer) defined as below:

$$q(Y(n)) = \begin{cases} 1, & Y(n) \geq T_n \\ 0, & Y(n) < T_n \end{cases}$$  \hspace{1cm} (3)

Based on previous experimental results\(^{25}\), $T_n$ 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 $V$:

$$V = \{v_i = (dx_i, dy_i), i = 0, \ldots, K\},$$  \hspace{1cm} (4)

Where $K$ is the number of neighbors being considered in the description. For example, if the center of patch and its four neighbors are considered in the description, the set $V$ is:

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

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

The description for a patch $Y_m$ is calculated as:

$$D(Y_m) = \sum_{i=0}^{K} q(Y(m + v_i)).$$  \hspace{1cm} (4)

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

Based on the results of previous study\(^{(25)}\), 10 different SBP neighborhood configurations from 10 to 28 neighbors are tested in this study. Fig. 1 shows the relative positions of all the SBP’s being tested.

### 2.3 Distance-Based Pre-Termination and Proposed Algorithm

For our NLM algorithm, patches in the search region are excluded from the weight calculation in two stages. The first stage compares the SBP’s of two patches. If the descriptions are different, the patch is skipped. Otherwise, the second stage is started to calculate the accumulated distance between \(Y_m\) and \(Y_n\) by sequentially subtracting corresponding pixels between the two patches. When the accumulated distance is larger than a threshold, the calculation of the accumulated distance is stopped and the patch is skipped. Given that the weight is equal to \(\exp(-d^2/h^2)\), the distance threshold is set as \(p_fh^2\) where \(p_f\) is a preset constant called “threshold factor”. In other words, only patches that satisfy \(D(Y_m) = D(Y_n)\) and \(\|Y_n - Y_m\|^2 \leq p_fh^2\) are included in the denoising calculation of Eq. (2).

### 3. Experimental Results

Our accelerated NLM algorithm was compared with the original NLM algorithm\(^{(1,2)}\). For all algorithms, \(h\) is set to \(100\sigma^2\). The mean of a patch is calculated over a 7x7 neighborhood. Four images from USC-SIPI image database: Baboon, F16, House, and Lena, were tested under the platform of a personal computer with 3.6 GHz Intel Core i7-77000 CPU and 8 GB of memory. All algorithms are implemented in C under Microsoft Visual Studio Community 2017.

Fig. 2 shows peak signal-to-noise ratio (PSNR) and average execution time vs. the number of neighborhood pixels in SBP. For PSNR, we can see that it generally increases and, then, slightly decreases as the number of neighborhood pixels in SBP increases. Since PSNR either reaches its peak or very close to the peak at 20 neighborhood pixels, we choose 20-neighbor SBP as the description for the first-stage classification. For average execution time, the original NLM algorithm and 20-neighbor SBP take 18.88 and 4.40 seconds to process an image, respectively. This translates to 77% reduction in execution time for 20-neighbor SBP. In other words, the execution time of 20-neighbor SBP.
neighbor SBP classification is only 23% of the original NLM algorithm. Although increasing the number of neighborhood pixels to 28 can further reduce the execution time by 2.5%, the potential reduction in image quality does not warrant such a small gain in execution time.

For distance-based pre-termination, threshold factors \( p_f \) from 1.0 to 3.0 are tested. Fig. 3 shows PSNR and average execution time vs. threshold factor \( p_f \). From the figures, we can see that PSNR generally increase and, then, reaches a plateau as \( p_f \) increases. Considering all the cases, we select \( p_f = 2.2 \) as the best threshold factor because PSNR reaches its maximum or very close to the maximum at this value. The average execution time for \( p_f = 2.2 \) is 4.01 seconds which is a further 9% reduction from that of the 20-neighbor-SBP pre-classified NLM.

Fig. 4 shows the comparison of PSNR between the original NLM and the proposed algorithm using 20-neighbor SBP and threshold factor \( p_f = 2.2 \). Comparing with the original NLM, the proposed method is much better than the original NLM except for the image “lena”. Between the proposed method and the original NLM, the maximum gap of PSNR is 2.37 dB at \( \sigma = 20 \) for the image “baboon”, 2.12 dB at \( \sigma = 40 \) for the image “f16”, 2.04 dB at \( \sigma = 30 \) for the image “house”, and 0.38 dB at \( \sigma = 20 \) for the image “lena”. Comparing with the SBP pre-classified method, the proposed method is either better or close to the SBP pre-classified method except for the image “f16”.

Fig. 5 show the images generated by the original NLM, the 20-neighbor SBP pre-classified method and the proposed method at \( \sigma = 30 \). For all the images, the proposed method...
shows much better perceptual quality than the original NLM even for the “lena” image where PSNR does not show significant improvement. There are no visual differences between the proposed method and the 20-neighbor SBP pre-classified method.

Since the purpose of distance-based pre-termination is to filter out the residuals that are left by SBP pre-classification, we can conclude that the distance-based pre-termination is indeed capable of reducing the computational complexity while maintaining the denoising performance. Therefore, distance-based pre-termination combining with SBP pre-classification is a good way to further increase the processing speed without sacrificing denoising performance of NLM.

4. Conclusion

In this paper, two approaches to reject dissimilar patches in NLM: simple binary pattern (SBP) pre-classification and early pre-termination based on patch distance have been studied through extensive experiments. Experimental results show that combining these two approaches can significantly reduce the execution time and effectively improve the denoising performance of NLM.

Acknowledgment

This work was supported by Ministry of Science and Technology, Taiwan, Republic of China, under contract MOST 106-2221-E-005-018.
Fig. 4. PSNR vs. the standard deviation of the noise $\sigma$ for 4 tested images: (a) baboon, (b) f16, (c) house, and (d) lena.

References

Fig. 4. Images generated by different NLM algorithms. The images from left to right are the input image, the original NLM, the 20-neighbor SBP pre-classified method, and the proposed method: (a) Baboon, (b) F-16, (c) House, and (d) Lena.


