FAST NON-LOCAL ALGORITHM FOR IMAGE DENOISING

Jin Wang[1,2], *Yanwen Guo*[2,3], *Yiting Ying*[2], *Yanli Liu*[2], *Qunsheng Peng*[2]

[1]Department of Computer Science, Xuzhou Normal University, Jiangsu, 221009, P.R China
[2]State Key Lab. of CAD&CG, Zhejiang University, Hangzhou, 310027, P.R China
[3]State Key Laboratory of Novel Software Technology, Nanjing University, Nanjing, 210093, P.R China
{jwang, ywguo, ytying, liuyangli, peng}@cad.zju.edu.cn

ABSTRACT

For the non-local denoising approach presented by Buades et al., remarkable denoising results are obtained at high expense of computational cost. In this paper, a new algorithm that reduces the computational cost for calculating the similarity of neighborhood windows is proposed. We first introduce an approximate measure about the similarity of neighborhood windows, then we use an efficient *Summed Square Image* (SSI) scheme and Fast Fourier Transform (FFT) to accelerate the calculation of this measure. Our algorithm is about fifty times faster than the original non-local algorithm both theoretically and experimentally, yet produces comparable results in terms of mean-squared error (MSE) and perceptual image quality.
Index Terms: Image enhancement, Wavelet transforms, Discrete Fourier transforms.

1. INTRODUCTION

Images are often corrupted by noise in acquisition and transmission, which usually degrades the quality of images. However, various image-related applications, such as aerospace, medical image analysis, object detection etc., generally require effective noise suppression to produce reliable results. Furthermore, denoising is often necessary as a pre-processing for other image/vision tasks, e.g. compression, segmentation and recognition. Therefore, denoising has been one of the most important and widely studied problems in image processing and computer vision.

The objective of denoising is to remove the noise effectively while preserving the original image details as much as possible. So far, many approaches have been proposed to get rid of noise. Traditionally, linear models, for example Gaussian filter [1], have been commonly used to reduce noise. These methods perform well in the flat regions of images. But a main drawback of them is that they are not able to preserve edges in a good manner. Edges, which are recognized as discontinuities in the image, are often smeared out. Nonlinear models however can handle edges in a much better way than those linear models. One popular model for nonlinear image denoising is the Total Variation (TV)-filter, introduced by Rudin et al. [2]. This filter is good at preserving edges, but tends to produce *mask* effect for smoothly varying regions in the input image. There are also a sort of denoising method, known as neighborhood filtering, which restores a pixel by taking an average of the values of its neighbors, e.g., bilateral filtering [3]. Employing the continuity of intensities for the pixels in a local neighborhood, those methods consider only the influence of the pixels in the neighborhood centered around it when adjusting a certain pixel.

In fact, there exist lots of repeat patterns in natural images. We can not only consider the local region when modifying a certain pixel, but also take the whole image into account. Therefore, Buades developed a non-local image denoising algorithm by making use of the information encoded in the whole image [4]. When modifying a pixel, the algorithm first computes the similarity between a fixed window centered around it and the windows centered around the other pixels in the whole image, then it takes the similarity as a weight to adjust this pixel. This method has shown remarkable and convincing results, but the efficiency is low for its pixel-wise window matching. The computational complexity of the original algorithm is $O(n^4)$, and a simplified algorithm present in their paper is about $49 * 441 * n^2$, in which n^2 is the number of pixels of the image. However, even with the simplified algorithm, it still takes about 1 minute to denoise a $640 * 480$ image on a common PC. In evidence, the high computational complexity makes it unfeasible to tackle with practical issues.

We present a denoising algorithm in the frame of non-local method by developing a fast calculation method for the comparison of windows' similarity. Exploiting a *Summed Squares Image* (*SSI*) scheme and Fast Fourier Transform (FFT), the per-pixel neighborhood matching is converted into the *SSI* pre-computing and efficient FFT. Computational complexity analysis and experiments indicate that our fast non-local algorithm is about 50 times faster than the original non-local algorithm, and yet produces similar results. Using our algo-

The work is supported by National 973 program (No.2002CB-312101), NSFC grant (No.60403038) and CUHK Direct Research Grant (No.2050349).

rithm, it normally takes less than 1 second to denoise a normal size image (e.g. $640 * 480$), and less than 10 seconds to denoise a 500 Megabyte photograph, which enables the algorithm widely applicable to practical situations.

The rest of this paper is organized as follows. In section 2, we introduce the fast non-local algorithm for image denoising. Section 3 analyzes our computational complexity and gives some experimental results. The last Section concludes the whole paper.

2. FAST NON-LOCAL DENOISING ALGORITHM

We first briefly introduce the non-local algorithm developed by Buades et al. [4]. Then we present our acceleration method based on *Summed Squared Image* (*SSI*) and fast Fourier transform (FFT), together with an approach for estimating the standard deviation of noise.

2.1. Non-local algorithm

In the non-local algorithm [4], given a noisy image,

$$I = \{I(x,y)|(x,y) \in \Omega\}, \tag{1}$$

the estimated value $I'(x_i, y_i)$ for a pixel (x_i, y_i) is computed as a weighted average of all the pixels in the image,

$$I'(x_i,y_i) = \sum_{(x_j,y_j)\in\Omega} w(i,j)I(x_j,y_j), \tag{2}$$

where the weight $w(i,j)$ of two pixels (x_i, y_i) and (x_j, y_j) depends on their similarity defined as,

$$w(i,j) = \frac{1}{Z(i)} e^{-\frac{S(i,j)}{h^2}}. \tag{3}$$

Here $Z(i)$ is the normalized constant, h is a constant proportional to the noise deviation σ^2. $S(i,j)$ is estimated by the weighted Euclidean distance of the two pixels' neighborhoods N_i and N_j with equal size (M, M) as:

$$S(i,j) = \|N_i - N_j\|_a^2, \tag{4}$$

where a is the standard deviation of the Gaussian kernel.

If n^2 is the number of pixels in the image, the complexity of this algorithm is $M^2 \cdot n^4$. By restricting the searching of similar window within the size of $21 * 21$ pixels and the neighborhood size $7 * 7$, the final complexity of the algorithm is $49 * 441 * n^2$. Obviously, high computational cost disallows it to be widely used in application.

2.2. The acceleration of non-local algorithm

For the convenience of acceleration, we adopt the Euclidean distance to compare two neighborhoods,

$$\begin{aligned} S(i,j) &= \|N_i - N_j\|^2, \\ &= \sum_{l=0}^{M-1}\sum_{m=0}^{M-1}[I_i(l,m) - I_j(l,m)]^2, \end{aligned} \tag{5}$$

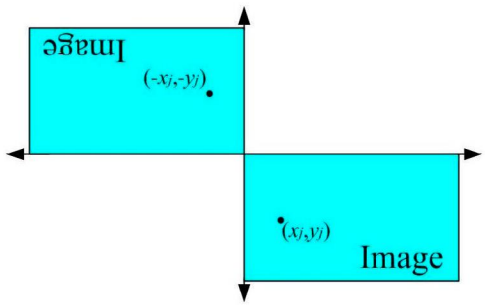

Fig. 1. *mirrored image*.

where $I_i(l,m)$ and $I_j(l,m)$ represent the corresponding pixels in N_i and N_j respectively

In fact, $I_j(l,m)$ in equ. (5) can be represented in the global coordinates on the mirrored image as: $I_j(l - x'_j, m - y'_j)$, with $x'_j = 3M/2 + x_j$, $y'_j = 3M/2 + y_j$ (see Fig. 1). So equation (4) is transformed into:

$$\begin{aligned} S(i,j) &= \sum_{l=0}^{M-1}\sum_{m=0}^{M-1}[I_i(l,m) - I_j(l - x'_j, m - y'_j)]^2 \\ &= N_i^2 + N_j^2 - N_i * N_j, \end{aligned} \tag{6}$$

where $N_i^2 = \sum_{l=0}^{M-1}\sum_{m=0}^{M-1}(I_i(l,m))^2$,
$N_j^2 = \sum_{l=0}^{M-1}\sum_{m=0}^{M-1}(I_j(l - x'_j, m - y'_j))^2$,
and $N_i * N_j = 2\sum_{l=0}^{M-1}\sum_{m=0}^{M-1}(I_i(l,m) \cdot I_j(l - x'_j, m - y'_j))$ denotes the convolution between N_i and N_j.

In above formula, $N_i * N_j$ can be figured out quickly with multiplifications under the fast Fourier transform, while N_i^2 and N_j^2 can be fast calculated as well using the *Summed Squared Image* (*SSI*) we propose in *subsection 2.3*. Note that if the compare window size is $M * M$, in (4), computing the similarity of the two compare window requires M^2 pixel operations, while, in our algorithm, it is figured out once which is achieved by means of FFT.

2.3. Summed Square Image (SSI)

The principle of *SSI* resembles *Integral image* which has been used in face detection [5]. For each pixel in the image, integral image maintains the summed value of all the pixels in the upper left part of the original image. Here we extend it to our *SSI*. Similar to the definition of integral image, for each pixel (x_0, y_0), *SSI* stores its sum for the squared values of the upper left pixels,

$$SSI(x_0,y_0) = \sum_{x\le x_0, y\le y_0} I^2(x,y). \tag{7}$$

SSI can be obtained in linear time proportional to the image size, we take the following algorithm to calculate it efficiently:
For $x_0 = 0, y_0 = 0$,

$$SSI(0,0) = I^2(0,0); \tag{8}$$

for $x_0 > 0, y_0 = 0$,

$$SSI(x_0, 0) = SSI(x_0 - 1, 0) + I^2(x_0, 0); \quad (9)$$

for $x_0 = 0, y_0 > 0$,

$$SSI(0, y_0) = SSI(0, y_0 - 1) + I^2(0, y_0); \quad (10)$$

for $x_0 > 0, y_0 > 0$,

$$\begin{aligned} SSI(x_0, y_0) &= SSI(x_0 - 1, y_0) + SSI(x_0, y_0 - 1) - \\ & SSI(x_0 - 1, y_0 - 1) + I^2(x_0, y_0). \end{aligned} \quad (11)$$

Obviously, with above algorithm, each pixel in the original image is processed only once, so the computational complexity for computing *SSI* is $O(n^2)$, in which n^2 is the size of the image.

By means of *SSI*, we can easily get the sum of squares for each pixel in any rectangles of the image within constant time. For example in Fig. 2, to calculate the sum of squares in rectangle D, only 3 addition operations are required,

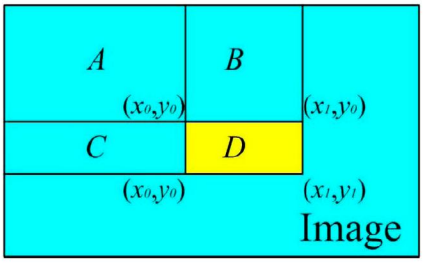

Fig. 2. *Using SSI to compute the summed squared pixels in the rectangle D.*

$$\begin{aligned} S_D &= S_{A \cup B \cup C \cup D} + S_A - S_{A \cup C} - S_{A \cup B} \\ &= SSI(x_1, y_1) + SSI(x_0, y_0) \\ &\quad -SSI(x_0, y_1) - SSI(x_1, y_0) \end{aligned} \quad (12)$$

Therefore, N_i^2 and N_j^2 can be computed quickly with equ. (11) once we get *SSI* of the noised image. Accurate analysis in section 3 will show that our fast non-local algorithm is much faster than the original one.

3. DISCUSSIONS AND EXPERIMENTATIONS

The most time-consuming part of the non-local algorithm [4] is the calculation of the Euclidian distance between similar windows in the image. For each pixel in the image, it takes $M^2 \cdot n^2$ square calculations, and the whole computational complexity is $M^2 \cdot n^4$ (M^2 denotes the size of similar windows and n^2 represents the number of pixels in the noised image).

In our fast non-local algorithm, this process has been transformed to computing convolution and summation of squares. It is well known that convolution becomes multiplication under Fourier transform, thus only n^2 multiplications need to be taken for each pixel in the image in our algorithm. Furthermore, Fourier transform can be performed quickly by modern hardware FFT accelerator within a neglectable time, convolution can therefore be fast carried out. As for the summations of squares, only addition operations are needed in this process which can also be neglected compared to the multiplications. In (4), computing the similarity of the two compare window requires M^2 pixel operations, while, in our algorithm, it is figured out once which is achieved by means of FFT. Thus the total computational complexity is n^4, which is M^2 times faster than the original algorithm.

In theory, the neighborhood size M should be congruent to the size of the repeated patterns in the image. In order to achieve convincing results, M should be set larger in higher resolution images, which leads to a significant slowdown in the original algorithm but no performance changing in ours.

As suggested in Buades' paper [4], M is set to be 7, and a $21 * 21$ search window is used instead of the whole image. In such simplification, instead of requiring 49 pixel operations when computing the similarity of two windows in [4], we figure out the similarity once by FFT and *Summed Squares Image* (*SSI*), thus the complexity of our algorithm is $441 * n^2$. Clearly, comparing to the original $49 * 441 * n^2$, our algorithm is about fifty times faster. This makes the performance of our algorithm acceptable to common users as is demonstrated in Table 1. Experiments on a PC with a Pentium IV 2.4G CPU and 512M RAM demonstrate that it takes no more than 10 seconds to denoise a 500Megabyte photo using our algorithm, while the original non-local algorithm requires nearly 10 minutes.

Our denoising results are comparable to that of the original non-local algorithm in terms of mean squared error (MSE). Table 2 compares the mean squared error (MSE) for different standard deviation of the added noise between our accelerated method and the original algorithm, note that the slightly difference in MSE is due to the use of the Euclidean distance when comparing two neighborhood instead of a weighted Euclidean distance in original non-local algorithm [4] ; Fig. 3 visualizes the comparison. Fig. 4 compares the denoising results of our acceleration method with that of non-local [4] for noisy Lena in which the standard deviation of noise is 20; Fig. 5 gives our denoising result of Lena.

Table 1. Performance results

Image size	Original NL	Fast NL algorithm	Ratio
$512 * 512$	28.16 secs.	0.35 secs.	80.5
$1024 * 768$	85.45 secs.	1.44 secs.	58.6
$2592 * 1944$	551.1 secs.	9.55 secs.	57.7

Table 2. MSE comparison

Stand deviation σ	5	10	15	20	25
Our acceleration method	14.6	32	55	81	110
Original non-local	12	30	52	68	106

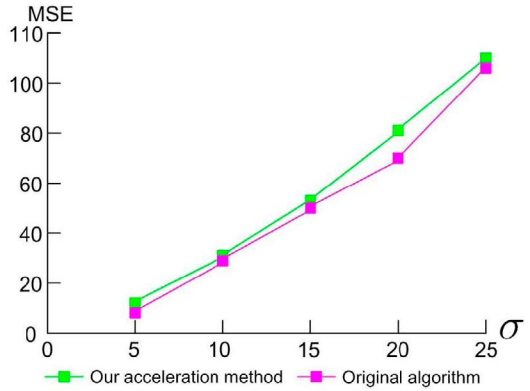

Fig. 3. *Comparison of the mean squared error (MSE) for different stand deviation of the noise between our acceleration method and original algorithm.*

Fig. 4. *Comparison between non-local effect and our acceleration result. From left to right, original image, noised image, the result of non-local [4] (MSE 68) and our result (MSE 81).*

4. CONCLUSIONS

Exploiting Summed Square Image (*SSI*) and fast Fourier transform (FFT), we proposed a fast non-local denoising algorithm in this paper. Theoretically and experimentally, the efficiency of our accelerated algorithm is about fifty times of the original algorithm, and the denoising results are still comparable to the results of the original algorithm both in MSE and perception. Thus our accelerated algorithm is feasible to tackle with practical problems.

5. REFERENCES

[1] M. Lindenbaum, M. Fischer, and A. Bruckstein, "On gabor contribution to image enhancement," *Pattern Recognition*, vol. 27, pp. 1–8, 1994.

[2] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259–268, 1992.

[3] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," *In: Proc. IEEE ICCV*, pp. 839–846, 1998.

[4] A. Buades, B. Coll, and J. M. Morel, "A non-local algorithm for image denoising," *In: Proc. IEEE CVPR*, vol. 2, pp. 60–65, 2005.

Fig. 5. *Our denoising result of noised lena (standard deviation 20).*

[5] P. Viola and J. Michael, "Rapid object detection using a boosted cascade of simple features," *In: Proc. IEEE CVPR*, 2001.