UDC 519.16
O. Pavliuk, R. Kutelmakh
Lviv Polytechnic National University
Software Engineering Department

INVESTIGATION OF EXISTING IMAGE DENOISING ALGORITHMS

The existing methods of image denoising as well as different noise types were investigated. Conclusions regarding the algorithms' quality, computational complexity and possible improvements were made.

Keywords: image algorithms, denoise, noise in digital signals, spatial domain, patches, frequency domain, wavelets

Introduction

Noise effect is typical problem for digital images. Its main reason is nonuniform photons distribution received by camera sensors [1]. Noise is random intensity variations or color information in image (which are absent in real image). For image *u* value of pixel *u(i)* is the result of light intensity measurement, made with the use of CCD matrix with light focusing system. Each CCD sensor counts an amount of input photons, observed during fixed time period. When light source is constant, an amount of photons received by each sensor ranges around its average value due to the Central limit theorem. Besides, each sensor, if not enough cooled, receives extra photons of warm.

Noise removal is very important task is many areas. It is used in computer tomography (to help doctor to recognise unwanted data on scan of an object), astronomical imaging, data transmission (to fix errors in received bits). There is high need of high-quality noise removal software in mobile devices and desktop platforms as well.

There are many noise removal methods, for example, patches methods [4,5] which focus on spatian domain, and wavelet methods which focuse on frequency domain [6-9]. Automatic metrics for quality estimation are considered. Existing noise removal methods compromise between high quality and low computational complexity. Non-Local Means algorithm provides high quality, while having quadratic complexity. Wavelet algorithms have near logarithmic complexity, suffering from local edge corrupting effects. It is important to develop an algorithm with computation complexity near to linearly-logarithmic which provides high quality of image restoration in close to real-time mode. This article is focused on comparing visual quality (such as edge preserving and blur effect) of existing methods for noise removal in real-life photos that are common for the users of mobile and desktop platforms.

Problem formulation

Generally image noise can be represented as:

$$v(i)=u(i)+n(i) ,$$

where *i* is image pixel, *v(i)* is observed value, *u(i)* is real value of *i*, which would be calulated as an average value of photons count, observed during a long period of time, and *n(i)* is noise value. The amount of noise depends on signal, thus the more is "real" value of *u(i),* the more is noise value *n(i)*. In existing noise models, normalized values of *n(i)* and *n(j)* at different pixels are considered as independent random variables, while gaussian noise itself as white noise [3].

All known noise removal algorithms depend on filtering parameter *h*. This parameter stands for filtering degree applied to the image. For most algorithms this parameter depends on the estimate of standard deviation σ [3]. Noise removal algorithm has to calculate "real" image *u(i)*, considering an image source and filtering parameter *h*. Types of image noise are described in the table 1.

Table 1

Noise type	Properties
Gaussian noise	• caused by pure lightening or high temperature • additive, independent for each pixel • in color cameras, where the blue channel has higher weight, it is more affected with noise
Salt-and-pepper noise	• "inpulse" noise • dark pixels in light regions of image and light pixels in dark regions of image • caused by errors in analog-to-digital conversion, bit transmission errors etc
Shot noise	• has Poisson distributution, which, except of very low intensity levels, is approximately Gaussian distribution • caused by statistical varations in photons count at current exposure level
Quantization noise	• caused by pixel quantization of input image into given count of levels (e.g. 255) • is uniformly distributed
Film grain	• has probability distribution near to shot noise • if film grains are uniformly distributed and each grain has equal probability of darkening after receiving photons, then number of such dark grains on image will be random and binomially distributed

Existing methods of noise removal

Noise removal methods are divided into two groups depending on the image domain in which they work:

1) patch-based algorithms (remove noise in spatial domain) [4,5];

2) wavelet-based algorithms (remove noise in frequency domain) [6,7,9].

For quality estimation of restored image for noise removal algorithms visual estimate is used as well as automatic metrics. The most used characteristic is Peak Signal To Noise Ratio (PSNR) [3]. An estimate of developed algorithm quality is obtained by getting its PSNR value for test image sets and then comparing results with existing algorithms. Final conclusion about the quality of developed algorithm is based on visual estimate.

PSNR characteristic

PSNR is ratio between maximum possible power and corrupting noise that affect an image. PSNR is usually expressed in decibel scale. PSNR is used mostly as quality characteristic of restored image. In such case the signal is input data and noise is an observation error. High PSNR value means high quality of output image.

For 8-bit one-channel image PSNR is calculated via mean square error (MSE):

$$MSE = \frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} [f(m,n) - f'(m,n)]^2$$

$$PSNR = 10 * \log_{10} \frac{255^2}{MSE}$$

Here 255 is maximum pixel value, MSE stands for mean square distance between real and observed images.

Patch-based algorithms

Patch is rectangular region of image, which considers local characteristics of given image pixel.

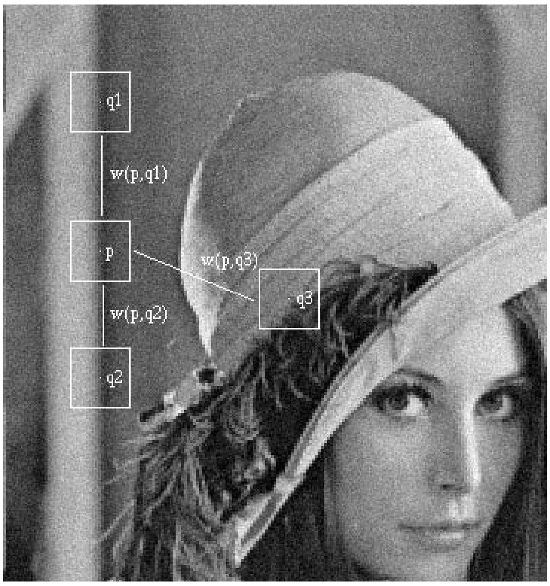

Fig. 1. Using pathes to compare pixels [5]

Main idea of patch-based algorithms [5] is to set new denoised pixel value as a sum of weighted average of similar image patches, with convolution filter applied, e.g. Gaussian blur. Typical patch-based algorithm is Non-Local Means (NLM) [5], which provides the best quality among known algorithms in spatial domain. Formula for calculating the value of pixel inside the patch can be expressed as:

$$u_i(p)=\frac{1}{C(p)}\sum_{q\in B(p,r)} u_i(q)*w(p,q)$$
,

where $i=1, 2, 3$ and $B(p, r)$ is square region neighbourhood, the center of which is pixel p, and size is $2r+1 \times 2r+1$ pixels. This search region is bounded with window of fixed size because of computational complexity constraints. The window has size 21x21 for small and medium values of noise σ. For large values of σ window size is increased to *35x35* to find more similar pixels.

The weight $w(p,q)$ depends on quadratic euclidian distance $d^2 = d^2(B(p,f), B(q,f))$ for $2f+1 \times 2f+1$ color patches, with centers p and q respectively.

$$d^2(B(p,f),B(q,f))=\frac{1}{3(2f+1)^2}\sum_{i=1}^{3}\sum_{j\in B(0,f)}(u_i(p+j)-u_i(q+j))^2$$

The restored value of each pixel is calculated with use of neighbouring (by value, not by distance) pixels. For each pixel new value will be a weighted average of similar pixels. To calculate the weights $w(p, q)$ exponential kernel is used:

$$w(p,q)=e^{-\frac{max(d^2-2\sigma^2,0.0)}{h^2}}$$

where σ is noise standard deviation and h is filtering parameter, based on the value of σ. Weight function is choosen in such a way to average the values of similar patches. Thus the weight of patches with distance less than $2\sigma^2$ is set to 1, while the weight of patches with larger distance is reduced.

Computational complexity of NLM is $O(n^2w^2p^2)$, where n is image size in pixels, w is search window size and p is patch size. Such complexity is quadratic, thus non-acceptable for large images. There are few improvements of NLM. One of them uses Principal Component Analysis [4] to get a vector of main patch characteristics, thus significantly reducing the complexity of calculating distances between patches, but reducing the quality of such distance calculation.

An effective algorithm which improves Non-Local Means builds cluster tree for patch space [4], thus logarithmically reducing an algorithm computational complexity. K-means clustering algorithm is used to divide patch space.

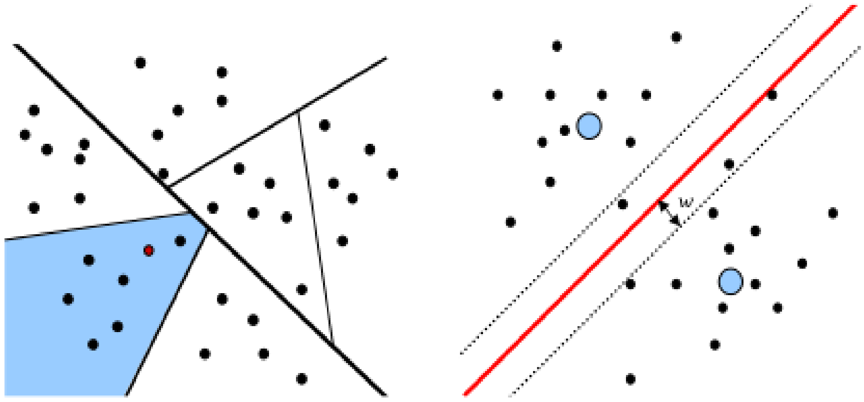

Fig. 2. K-means algorithm for patch cluster tree

Patch space is recursively divided into 2 parts, resulting in binary cluster tree, which is used for similar patches search. PCA characteristics are used to compare two patches. The stop condition of recursion is satisfied when number of patches at some level of tree is less than given threshold (in [4] this threshold is equal to 30).

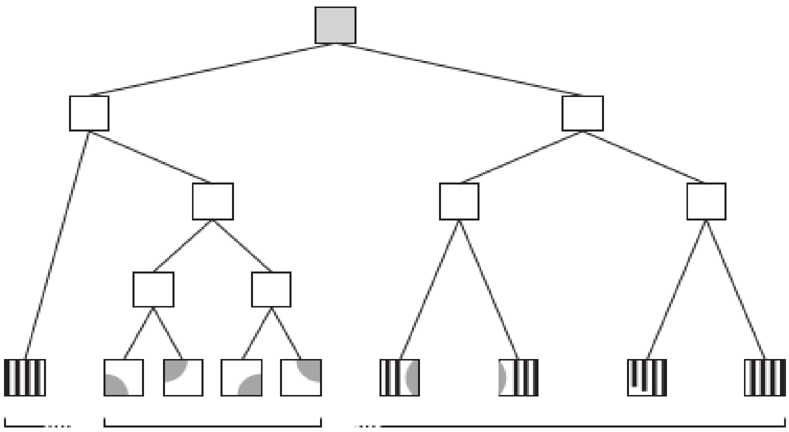

Fig. 3. Cluster tree of patches

The last level of cluster tree is considered. The new value of pixel u_i (p) is calculated as:

$$u_i(p)=\frac{1}{C(p)}\sum_{q\in B(p,r)} u_i(q)*w(p,q)$$
,

where $B(p, r)$ is patch space at some leaf of cluster tree.

Computational complexity of this algorithm is $O(n*log(n))$ where $n=r_w^2$ is the size of patch space.

Wavelet-based algorithms

The idea of digital wavelet transform is that signal is divided into two subbands with high and low frequencies. Noise is left in sigh signal subband, threshold function is applied to it [7]. Wavelets are functions which satisfy certain mathematical conditions [6]. The name “wavelet” implies that function should be zero-intergrated, “waving” around x-axis. Wavelet function should be well-localized. Other requirements are technical and are needed for providing fast and exact calculation of direct and inverse wavelet-transforms. Many types of wavelets are developed: smooth wavelets, compactly supported wavelets, wavelets with simple mathematical expressions etc. The most famous examples are Haar and Daubechi wavelets. Like sines and cosines in Fourier Transform, wavelets are used to represent other functions [7-9]. Wavelet function is called “mother wavelet” and expressed with formula:

$$\psi_{s,u}(t)=\frac{1}{\sqrt{(s)}}\psi\frac{(t-u)}{s}$$

, where s is scale and u is time shift.

Continuos wavelet transform of function *f(t)* is calculated as convolution of input signal with mother wavelet (fig. 4).

$$W\ f(s,u)=\int_{-\infty}^{\infty} f(t)\,\psi_{s,u}(t)\,dt$$

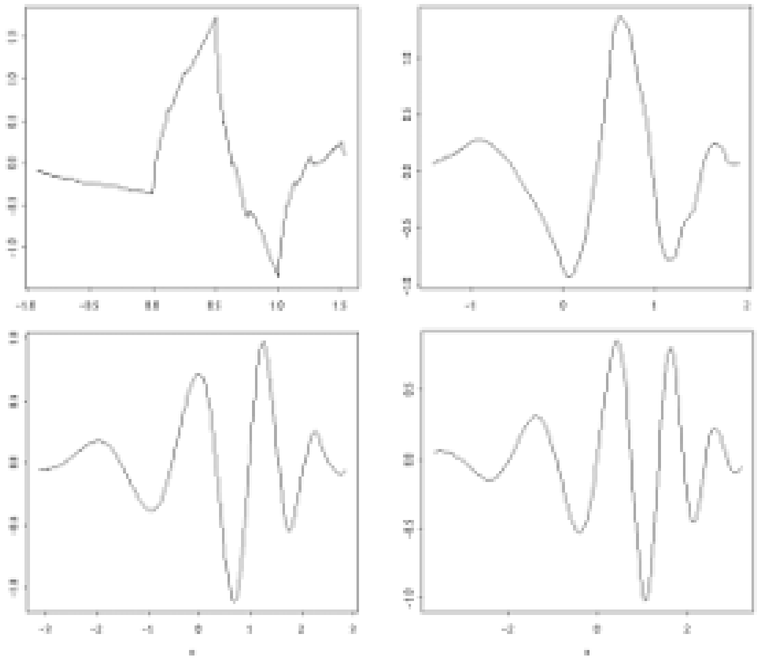

Fig. 4. Daubechi wavelets Ψ(u)

Digital wavelet algorithm for image data uses One-dimensional Digital Wavelet Transform (1D DWT) by rows followed 1D DWT by columns.

1D DWT algorithm is performed in such steps:

1. Subband decomposition. For input data signal *y[i]* of size *n* apply high-frequency and low-frequency wavelet filters thus getting two signal subbands of half-size *n/2*. For Daubechi wavelet, which has coefficients D_4 = *[0.482962, 0.836516, 0.224143, -0.129409]*, the formula for high-frequency part is:
   *high[v] = y[2*v]*D₄[0] + y[2*v+1]*D₄[1] + y[2*v+2]*D₄[2] + y[2*v+3]*D₄[3],*
   while the low-frequency part is calculated using formula:
   *low[v] = y[2*v]*D₄[3] - y[2*v+1]*D₄[2] + y[2*v+2]*D₄[1] - y[2*v+3]*D₄[0].*
2. Subband decomposition (step 1) is recursively repeating for each subband to the level, set up as parameter. Scheme of wavelet coefficients is given on figure 5, where at the top left corner of image high frequencies are localised.
3. Wavelet coefficients with absolute value less then given threshold σ are set to zero.
4. Inverse Digital Wavelet Transform is applied to received coefficients thus giving the resulting image.

Computational complexity of this algorithm is *O(n*log(n))* where *n* is the size of subband at each wavelet level.

Fig. 5. Wavelet coefficients at the 4th level of signal decomposition

Visual comparison of investigated algorithms

The quality of existing algorithms' results on real-life images was investigated. An image with additive white Gaussian noise $\sigma = 0.1$ of size 400x600 pixels (fig. 6) was given as input to existing denoise algorithms. Perceptual quality was compared for Non-Local Means method, the Cluster Tree modification of NLM, and Digital Wavelet Transform with thresholding. Figures 7-9 show the results.

Fig. 6. Noisy image

Fig. 7. Result, obtained with Non-Local Means method

Fig. 8. Result, obtained with Cluster Tree NLM modification

Fig. 9. Result, obtained with Digital Wavelet Transform

Non-Local Means provides good quality, but high computational complexity. Cluster Tree modification gives worse results while having lower computational complexity. Wavelet transform gives acceptable results with unwanted local blur artifacts at the edges.

Conclusions

Visual quality of existing denoise algorithms on real-life images was compared. Patch-based algorithms provide good visual quality but have high computational complexity due to the large size of patch space. PCA is used and then patch cluster tree is built to decrease the size of patch space thus reducing the computational complexity. The existing implementations of Non Local Means work slowly, thus their use is impossible in real time on mobile or desktop platforms. There is a need to develop an algorithm, which works with patches using cluster tree.

The algorithms which use wavelet-transform, have less computational complexity (*O(n*log(n))*), but provide unwanted artifacts at the edges of objects. It is reasonable to develop an algorithm which uses wavelet transform and choose such wavelet functions that give the best visual quality of restored image.

The investigation performed gives the ability to estimate the visual quality of noise removal algorithms. It is reasonable to use patch algorithms for images with high standard deviation ($\sigma > 2.5$), while for uniform images it is reasonable to choose image specific-aware wavelet function and then apply wavelet transform with thresholing of wavelet coefficients. In is planned to develop new universal high quality algorithms with close to linear-logarithmic computational complexity, which combine both of the existing noise removal methodologies.

1. O. Schwartz and E. P. Simoncelli, "Natural signal statistics and sensory gain control," Nature Neurosci., vol. 4, no. 8, pp. 819–825, Aug. 2001. 2. R. Kimmel, "Demosaicing: Image reconstruction from CCD samples," IEEE Trans. Image Process., vol. 8, no. 9, pp. 1221–1228, Sep. 1999. 3. Z. Wang, H. Sheikh, and A. Bovik. No-reference perceptual quality assessment of JPEG compressed images. In ICIP, pages I–477 – I–480 vol.1, 2002. 4. Charles-Alban Deledalle, Joseph Salmon, Arnak Dalalyan (2008), Image denoising with patch based PCA: local versus global: Université Paris Diderot-Paris, Paris, France. 5. Thomas Brox, Oliver Kleinschmidt, Daniel Cremers (2008), Efficient Nonlocal Means for Denoising of Textural Patterns. Saarland University, Saarbr¨ucken, Germany. 6. S. Mallat, A Wavelet Tour of Signal Processing. New York: Academic, 1998. 7. S. G. Chang, B. Yu, and M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression," IEEE Trans. Image Process., vol. 9, no. 9, pp. 1532–1546, Sep. 2000. 8. I. Daubechies, "Orthonormal bases of compactly supported bases," 708 Commun. Pure Appl. Math., vol. 41, pp. 909–996, Dec. 1988. 9. R. R. Coifman and D. L. Donoho, Wavelets and Statistics, A. Antoniadis and G. Oppenheim, New-York: Springer-Verlag, 1995.