ПРОГРАММИРОВАНИЕ, 2014, № 2, с. 1–9

УДК 004.421.6

HeNLM-LA: A LOCALLY ADAPTIVE HERMITE FUNCTIONS EXPANSION BASED NON-LOCAL MEANS DENOISING

N. V. Mamaev, A. S. Lukin, D. V. Yurin

The Faculty of Computational Mathematics and Cybernetics
Lomonosov Moscow State University
119991 Moscow, Leninskie Gory, 1
E-mail: mamaev.nikolay93@mail.ru, lukin@ixbt.com, yurin@cs.msu.su

Поступила в редакцию 13.01.2014

A new noise reduction algorithm HeNLM-LA is proposed. It is a modification of the non-local means algorithm using Hermite functions expansion of pixel neighborhoods. The filtering strength parameter is automatically adjusted proportionally to the local noise level. An algorithm for local noise level estimation is based on edge modeling; it suppresses high-amplitude edges in the map of local image variance.

1. INTRODUCTION

Image denoising problem is important in many fields of image processing. One popular class of image denoising algorithms is averaging of pixels, where pixel weights depend upon similarity of some values characterizing pixel neighborhoods [1], [2], [3]. The first algorithm, known as *Non-Local Means* (NLM), calculates weights depending on Euclidean distance between whole blocks (patches) around respective pixels. Algorithms [2], [3] are modifications of NLM and allow to increase the effectiveness of noise reduction. They are based on replacing the patches by feature vectors characterizing them. The cardinality of feature vectors is much lower than the number of pixels in a patch, which allows to reduce the computational complexity. GFNLM algorithm [3] is based on features obtained by expansion of the neighborhood into Gabor functions. In the basis of LJNLM-UR algorithm [2] lies the Taylor series expansion of the neighborhood. Coefficients of the expansion (known as *Local Jets*) are defined by derivatives of the Gaussian function. The advantage of [2] is that the search for similar patches does not only consider parallel shifts, but also rotations by an arbitrary angle.

Many image denoising algorithms significantly depend on a parameter characterizing the noise level in the image. For ε_V [4], bilateral filtering [5] and Non-Local Means algorithms [1], [2], [3] the parameter has a meaning of a range (or standard deviation) of brightness values which can be considered as noise and suppressed. The natural approach is to set the parameter proportionally to the noise standard deviation (assuming that the mean value of noise is zero). However the noise level in real images is usually unknown and requires estimation.

When it is known that the noise level is constant across the whole image, the typical approach is to scan the image by a small window, computing the variance of pixel values in the window and choosing the estimated noise level as the minimal variance value among all possible positions of the window. This approach is based on the assumption that the image contains a fragment with a size larger than the window size which has the brightness close to constant. An example of such approach is described in [6] where the gradient value is being averaged in texture-free areas of the image. Also [6] considers the case when noise level depends on intensity of the image—such a dependency is observed when images are obtained from the camera with gamma correction. The future development of [6] is described in [7] where a robust probabilistic noise level estimator depending on brightness and position on the image is constructed.

A significant problem of methods based on a search for the minimum of variance is existence of edges and textures on images, and absence of flat areas [8]. These image features lead to overestimation of noise level in most algorithms, including the popular estimate based on a median of high-frequency wavelet coefficients. In [8] an approach based on decomposition of image blocks by principal component analysis is proposed. Here the minimal eigenvalue of the covariance matrix is chosen as the noise level estimate. This allows to obtain the noise estimate even in the areas with regular texture, if the texture is "predictable" (is well approximated by the PCA).

In this article a new method of non-local filtering, HeNLM, is proposed. It is based on pixel block expansion in Hermite functions. Inheriting all advantages of [2] approach, HeNLM better distinguishes textures due to higher independence of feature vector components because of the orthogonality of Hermite functions, and better description of high-frequency components of the local neighborhood. The second contribution of this article is a new method of local noise estimation giving rise to the locally adaptive version of the HeNLM-LA algorithm. The proposed method of noise estimation is algorithmically and computationally simpler than [7], [8], but unlike [9] it does not lead to underestimation of the noise level.

2. LOCAL JETS BASED NON-LOCAL MEANS

In the Non-Local Means algorithm the value of the output pixel $f(x,y)$ is the weighted sum of values of source image pixels $I(x,y)$ from the neighborhood Q:

$$f(x,y) = \frac{1}{\sum\limits_{(\xi,\eta)\in Q} w} \sum_{(\xi,\eta)\in Q} w(x,y,\xi,\eta) I(x+\xi, y+\eta) \tag{2.1}$$

These weights $w(x,y,\xi,\eta)$ depend upon similarity of blocks $v(x,y)$ around pixels with coordinates (x,y)

$$w(x,y,\xi,\eta) = exp(-\frac{\|v(x,y) - v(x+\xi, y+\eta)\|_2^2}{2\rho^2}) \tag{2.2}$$

The advantage of this method is a high quality of the resulting image. However the NLM algorithm has high computational complexity. Also the original method does not consider rotation of blocks when calculating averaging weights [1], i.e. for pixels lying on one edge, but with different gradient directions, weights will be small (2.1). This can lead to poor noise reduction along edges where the gradient has a different direction in each pixel of the edge. Also the NLM algorithm is highly sensitive to noise because the computation of weights is performed by comparing raw (noisy) pixel values.

The Local Jets method (algorithms LJNLM-LR and LJNLM-UR [2]) partially overcomes the shortcomings of [1]. In LJNLM, weights for averaging pixels depend upon the distance between feature vectors characterizing pixel neighborhoods. Components of the feature vector are values of Taylor series expansion coefficients of the pixel neighborhood, which are determined by convolutions of the source image $I(x,y)$ with Gaussian derivatives at different scales:

$$f_{nm}^{\sigma} = I(x,y) * (\frac{\sigma^{n+m}}{1+n+m} \frac{d^{n+m} G_\sigma}{dx^n dy^m}) \tag{2.3}$$

The Gaussian function is given by

$$G_\sigma(x,y) = \frac{1}{2\pi\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}} \tag{2.4}$$

The multiplier σ^{n+m} is introduced in (2.3) for equal filter response to same textures at different scales [10]. In other words, if σ and the scale of $I(x,y)$ are increased synchronously, the values in (2.3) will not change. The denominator $1+n+m$ represents the number of derivatives of order $n+m$ and excludes the situation when the contribution of high-order derivatives leads to ignoring of low-order derivatives.

An advantage of the method is ease of obtaining features that are invariant to rotation. The features are rotated to the coordinate system (g,τ), where g is the gradient in pixel (x,y) and τ is perpendicular to g (fig. 1).

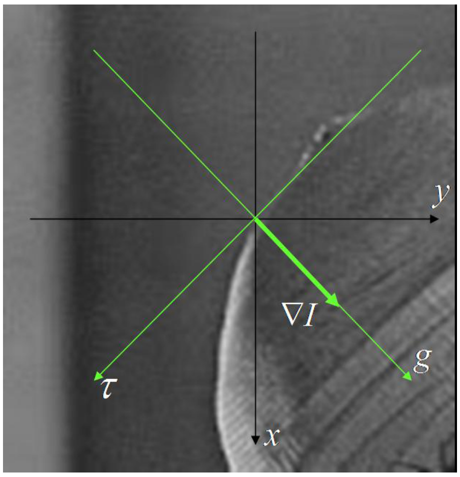


Fig. 1: Coordinate system (g, τ).

The feature vector characterizing the neighborhood of pixel is defined as follows:

$$P = \{\tilde{f}^{\sigma}_{nm}; n + m \le r; \sigma \in S\} \qquad (2.5)$$

where $\tilde{f}^{\sigma}_{nm}$ is a feature in the coordinate system (g, τ), r is the maximal order of derivatives, S is a set of different scales [10].

3. HERMITE FUNCTIONS

Hermite functions are defined as [12]:

$$\psi_n(x) = \frac{1}{c_n} e^{\frac{x^2}{2}} \frac{d^n(e^{-x^2})}{dx^n} = \frac{(-1)^n}{c_n} H_n(x) e^{\frac{-x^2}{2}},$$

$$\text{where } c_n = \sqrt{\sqrt{\pi} 2^n n!}$$

$$H_n(x) = (-1)^n \frac{d^n e^{-x^2}}{dx^n} e^{x^2} \text{ is a Hermite polynomial} \qquad (3.1)$$

They form the complete orthonormal system in $L_2(-\infty, +\infty)$ [11]:

$$\int_{-\infty}^{+\infty} \psi_n(x) \psi_m(x) dx = \delta_{nm} \qquad (3.2)$$

We define multiscale Hermite functions as follows:

$$\psi^{\sigma}_n = \frac{1}{\sigma} \psi_n(\frac{x}{\sigma}) \qquad (3.3)$$

The multiplier $\frac{1}{\sigma}$ in (3.3) provides that ψ^{σ}_n, unlike (3.2), are not normalized to one, but normalized by the filter response—see the comment after (2.4). Hermite functions (3.3) satisfy the differential equation [11]:

$$\psi^{\sigma}_n + (\sigma^2(2n + 1) - x^2)\psi^{\sigma}_n(x) = 0 \qquad (3.4)$$

Some of Hermite functions and Gaussian function derivatives are shown in fig. 2. It is seen that the localization (support) area of Hermite functions and Gaussian derivatives is roughly same but Hermite functions much better represent peripheral parts of the area, because their amplitudes stay roughly the same there.

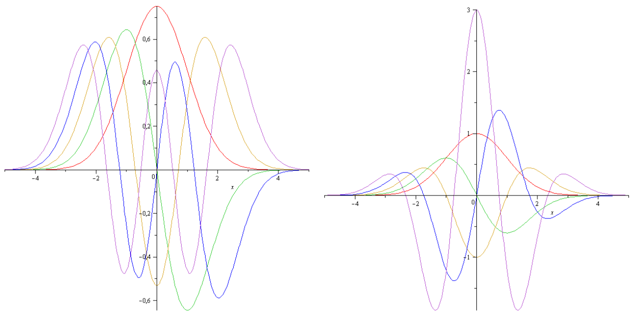


Fig. 2: Hermite functions (on the left) and Gaussian function derivatives (on the right).

We define 2D Hermite functions [12] as:

$$\psi^{\sigma}_{nm}(x, y) = \psi^{\sigma}_n(x) \psi^{\sigma}_m(y) \qquad (3.5)$$

4. HERMITE FUNCTIONS BASED NON-LOCAL MEANS

We propose a modification of the LJNLM-LR method using the expansion into Hermite functions instead of Gaussian derivatives. Now elements of the feature vector are convolutions of the source image u with Hermite functions:

$$u^{\sigma}_{nm} = u * \psi^{\sigma}_{nm} \qquad (4.1)$$

As well as in LJNLM-LR method, obtained features are converted in the coordinate system (g, τ). We derive an explicit expression. Consider two coordinate systems related by a rotation:

$$\begin{pmatrix} \xi \\ \eta \end{pmatrix} = R \begin{pmatrix} x \\ y \end{pmatrix}, R = \begin{pmatrix} \cos\vartheta & \sin\vartheta \\ -\sin\vartheta & \cos\vartheta \end{pmatrix} \qquad (4.2)$$

Then the differential operator in a new coordinate system can be expressed through the linear combination of operators in the old coordinate system:

$$\frac{d}{d\xi_i} = \sum_{j=1,2} R_{ij}\frac{d}{dx_j}, where \begin{matrix} i=1,2;\ \xi_1=\xi;\ \xi_2=\eta \\ j=1,2;\ x_1=x;\ x_2=y \end{matrix} \quad (4.3)$$

The expression for Gaussian derivatives takes the form:

$$\frac{d^{n+m}G_\sigma}{d\xi_1^n d\xi_2^m} = \left(\sum_{j=1,2} R_{1j}\frac{d}{dx_j}\right)^n \left(\sum_{j=1,2} R_{2j}\frac{d}{dx_j}\right)^m G_\sigma \quad (4.4)$$

The Hermite function is a product of a Gaussian derivative, a radially symmetric multiplier $e^{\frac{(x^2+y^2)}{2}}$, and a coefficient c_j. By substitution of (3.1) into (3.3) and (3.3) into (3.5), considering (4.4), removing brackets around Gaussian function derivatives and considering the form of the matrix R (4.2), it can be obtained that:

$$c_n c_m \psi^\sigma_{nm} = \sum_{j=0}^{n+m} a_j c_j c_{n+m-j}\psi^\sigma_{j,n+m-j},$$

$$a_j = \sum_{k=max(0,j-m)}^{min(j,n)} (-1)^{j-k}(cos\vartheta)^{m-j+2k}\cdot$$

$$\cdot (sin\vartheta)^{n+j-2k} C_n^k C_n^{j-k}$$

$$\text{where} \quad C_n^k = \frac{n!}{k!(n-k!)} \quad (4.5)$$

We note that the expression for rotation of vector components for the Local Jets method is the same, but coefficients c_j should be set equal to 1.

By choosing the new basis (ξ,η) so that the ξ axis direction matches the brightness gradient of the image $\vec{g} = \left(I(x,y) * \frac{dG_\sigma}{dx}, I(x,y) * \frac{dG_\sigma}{dy}\right)^T$ it can be obtained that in (4.2) and (4.5) $\frac{\vec{g}}{|\vec{g}|} = (\cos\vartheta, \sin\vartheta)^T$.

The use of Hermite functions instead of Gaussian derivatives better characterizes pixel neighborhoods, because their orthogonality means less dependence between feature vector components h^σ_{nm} (4.1) and, accordingly, their greater significance. Also the effective localization region is expanded and, accordingly, the peripheral data of the local neighborhood and high-frequency components are better represented.

5. NOISE LEVEL ESTIMATION FOR AUTOMATIC SELECTION OF THE FILTERING PARAMETER

Consider the model image of an ideal step edge with additive Gaussian noise:

$$I(x,y) = \begin{cases} a+\nu(x,y),\ x>0 \\ -a+\nu(x,y),\ x<0, \end{cases}$$
$$\langle \nu(x,y)=0\rangle, \langle \nu^2(x,y)\rangle = \beta^2 \quad (5.1)$$

where parameter a characterizes the brightness of the image and $\nu(x,y)$ is the additive noise. We will perform averaging with a Gaussian window, i.e.

$$\langle f(x,y)\rangle = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x-\xi,y-\eta)\frac{1}{2\pi\sigma^2}e^{-\frac{\xi^2+\eta^2}{2\sigma^2}}d\xi d\eta \quad (5.2)$$

We compute the local variance of the image (5.1):

$$D(x,y) = \langle I^2(x,y)\rangle - \langle I(x,y)\rangle^2 \quad (5.3)$$

Using averaging (5.2) and taking into account that the noise mean is zero, it can be obtained that:

$$\langle I(x,y)\rangle = \frac{1}{\sqrt{2\pi}\sigma}\left(a\int_{-\infty}^{x} e^{\frac{-\xi^2}{2\sigma^2}}d\xi - a\int_{x}^{+\infty} e^{\frac{-\xi^2}{2\sigma^2}}d\xi + \right.$$
$$\left. + \int_{-\infty}^{+\infty}\nu(\xi)e^{\frac{-\xi^2}{2\sigma^2}}d\xi\right) = a\cdot erf(\frac{x}{\sqrt{2}\sigma}) \quad (5.4)$$

By analogy:

$$\langle I^2(x,y)\rangle = \frac{1}{\sqrt{2\pi}\sigma}\left(a^2\int_{-\infty}^{+\infty} e^{\frac{-\xi^2}{2\sigma^2}}d\xi + \right.$$
$$+2a\int_{-\infty}^{x}\nu(\xi)e^{\frac{-\xi^2}{2\sigma^2}}d\xi - 2a\int_{x}^{+\infty}\nu(\xi)e^{\frac{-\xi^2}{2\sigma^2}}d\xi$$
$$\left. + \int_{-\infty}^{+\infty}\nu(\xi)e^{\frac{-\xi^2}{2\sigma^2}}d\xi\right) = a^2+\beta^2 \quad (5.5)$$

Here we extensively interpret the equality of the noise mean to zero (second and third integrals in brackets) by assuming that the integral of the noise component is equal to zero in a part of the neighborhood with Gaussian weights, or in any neighborhood with any weights if the area with nonzero weights is significantly wider than the noise correlation radius. If noise correlation radius is significantly smaller than σ and the noise distribution statistics is symmetric, this condition is satisfied. Then by substituting (5.4) and (5.5) into (5.3) the following expression for the local variance can be obtained:

$$D_\sigma(x,y) = \beta^2 + a^2(1 - erf^2(\frac{x}{\sqrt{2}\sigma})) \qquad (5.6)$$

It can be noted that the expression is not equal to the value of variance, but it contains an additive term which becomes significant near the edge (fig. 4).

On the other hand, edges in the image can be found using absolute brightness gradient, as in the Canny method [13]. For the model image (5.1) the gradient can be calculated as convolution with a Gaussian derivative, and again, assuming the equality of noise mean to zero, it follows that:

$$g_\sigma(x,y) = \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{+\infty} I(x-\xi)\frac{\xi}{\sigma^2}e^{\frac{-\xi^2}{2\sigma^2}}d\xi =$$

$$= -\frac{1}{\sqrt{2\pi}\sigma}\left(2ae^{\frac{-x^2}{x\sigma^2}} + \int_{-\infty}^{+\infty}\nu(\xi)\frac{\xi}{\sigma^2}e^{\frac{-\xi^2}{2\sigma^2}}d\xi\right) =$$

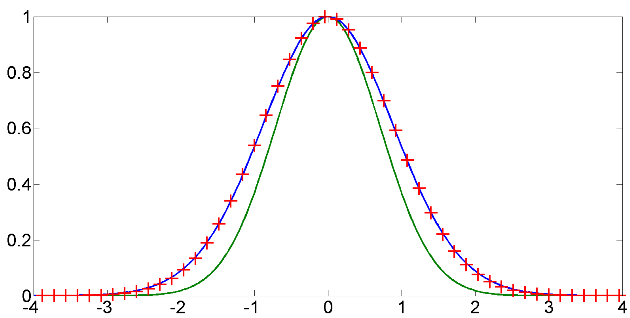
$$= \frac{\sqrt{2}a}{\sqrt{\pi}\sigma}e^{\frac{-x^2}{2\sigma^2}} \qquad (5.7)$$

and the square of gradient modulus:

$$g_\sigma^2(x,y) = \frac{2a^2}{\pi\sigma^2}e^{\frac{-x^2}{\sigma^2}} = \frac{2a^2}{\pi\sigma^2}F_\sigma(x),$$

$$\text{where} \quad F_\sigma(x) = e^{\frac{-x^2}{\sigma^2}} \qquad (5.8)$$

Comparing expressions (5.6) and (5.8) we can see that far from the edge the gradient is equal to zero and the variance is equal to noise variance. Near the edge both functions experience “bumps” of similar shape: the expression from (5.6) has a shape similar to the Gaussian (5.8) with the appropriate choice of parameter σ_0 (fig. 3).



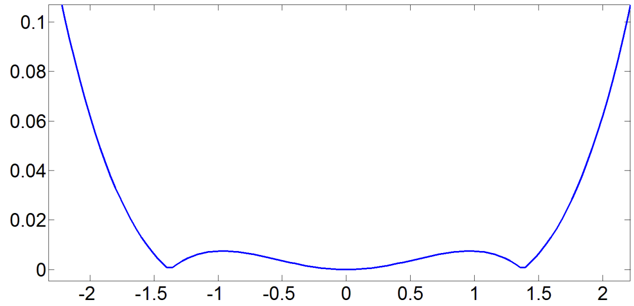


Fig. 3: Graph of functions $f(x) = 1 - erf^2(\frac{x}{\sqrt{2}\sigma})$ (wider), $\sigma = 1$, $F_1(x)$ (narrower), $F_{1.27}(x)$ (markers) and relative error of functions $f(x)$ and $F_{1.27}(x)$ $\left|\frac{f(x)-F_{1.27}(x)}{f(x)}\right|$ (below)

Consider the minimization problem:

$$\sigma_0 = \arg\min_\sigma \left(\int_{-\infty}^{+\infty}(F_{\sigma_0}(x) - f(x))^2dx\right)^{1/2} \qquad (5.9)$$

Its numerical solution has provided that the minimum is reached for $\sigma_0 \approx 1.27\sigma$. At the same time, the difference norm is equal to 0.009. So, the expression (5.6) for variance can be replaced with the approximation:

$$D_\sigma(x,y) = \beta^2 + a^2G_{\sigma_0}(x), \ \sigma_0 \approx 1.27\sigma \qquad (5.10)$$

Comparing (5.10) and (5.8) it can be seen that if the gradient modulus is calculated with a larger value $\sigma_0 \approx 1.27\sigma$, the expression (5.10) can be written as:

$$D_\sigma(x,y) = \beta^2 + \frac{\pi\sigma_0^2}{2}g_{\sigma_0}^2, \qquad (5.11)$$

And the noise level can be evaluated as

$$\beta^2(x,y) = D_\sigma(x,y) - \frac{\pi\sigma_0^2}{2}g_{\sigma_0}^2(x,y), \quad \sigma_0 \approx 1.27\sigma \qquad (5.12)$$

It is important to emphasize that expressions (5.6), (5.8) and (5.10) significantly reflect the form of the model image (5.1) due to spatially-dependent functions (Gaussian or error function), but the right part of expression (5.12) contains locally calculated (through convolutions) characteristics: the local variance $D_\sigma(x,y)$ and the square of the gradient modulus $g_{\sigma_0}^2(x,y)$. The noise variance is a linear combination of these values computed locally for each pixel with constant coefficients. Therefore results obtained for the model image can be applied to any image and obtained noise level estimates can be considered as local, depending on the coordinates. We also note that in the numerical implementation it is possible to calculate the gradient modulus with the same value of σ as the image variance (5.2), (5.3). However in this case before the substitution of the gradient modulus in (5.12) it will be necessary to blur it by convolution with a Gaussian function with $\sigma_3 = \sqrt{\sigma_0^2 - \sigma^2} \approx \sigma\sqrt{1.27^2 - 1} = 0.783\sigma$.

6. RESULTS

The results of applying the noise estimation algorithm to the digital image described by formula (5.1) are shown in fig. 4. It can be seen that the bright white stripe corresponding to the edge in the local variance image (4c) is absent in the image (4d). By averaging in the entire area of the image (4d) the mean estimated noise level of $\hat{\beta} = 0.13$ was obtained with original simulated noise level being $\beta = 0.14$.

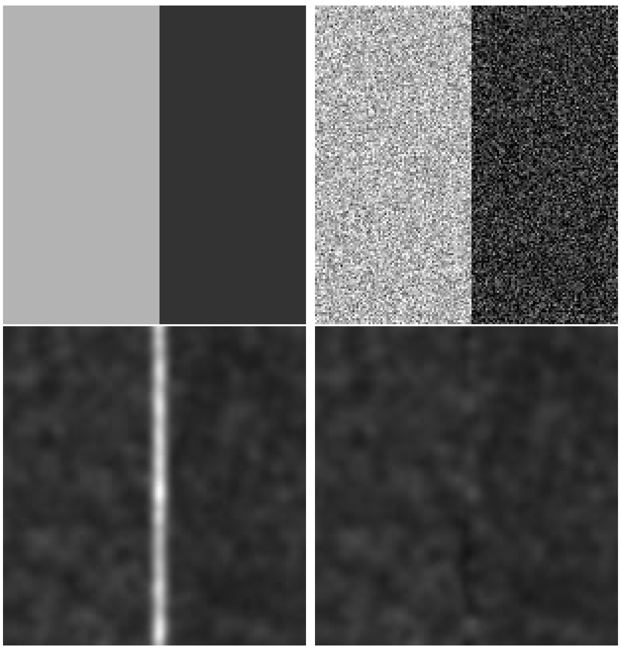


Fig. 4: Result of noise level estimation on the model image: source image; source image with Gaussian noise added; variance of the noisy image; result of noise estimation.

The results of processing of real images are shown in fig. 5. Gaussian noise was added to images *Lena* и *Barbara*.

It can be seen that in the first image *Lena* with high level of added noise the contours of objects are clearly visible in the second column and completely absent in the third one. With small noise (in the next row) the area with a complex texture (feathers on the hat) was estimated as noise and is shown with a brighter color in the third column. Edges of the objects are partially visible, often with darker color, i.e. the obtained level of noise is underestimated. We will comment on this later. In the third row (image *Barbara*) with low noise level it is clearly visible that textured areas (pants, scarf, tablecloth) are shown with a brighter color, i.e. the fine texture was recognized by the algorithm as noise, which does not contradict with the logic and the goal of the algorithm. The estimated level of noise generally corresponds to the added noise, see fig. 6. Our studies found that the optimal filtering parameter ρ^2 (the one that maximizes PSNR) and the mean estimated level of noise depend linearly. This is no linear dependence, however, at low noise levels, because textured areas in this case provide a larger contribution to the variance than the noise itself.

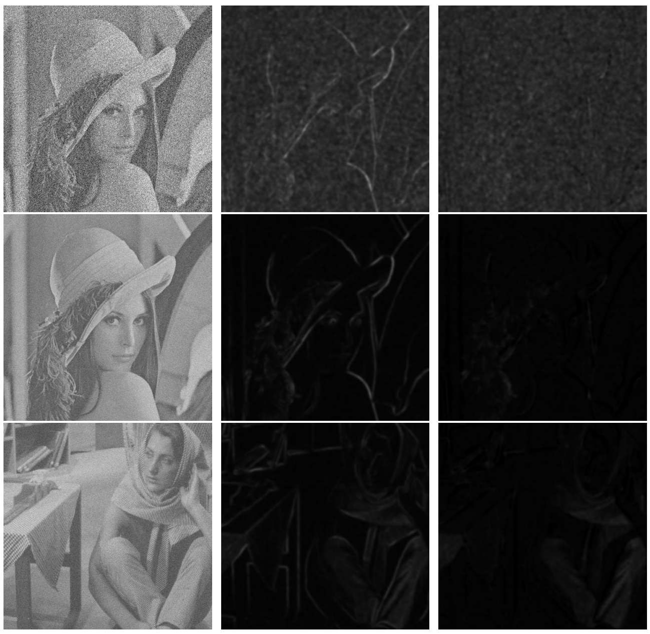


Fig. 5: Results of local noise estimation on real images. The first column is the source noisy image. The second column is the local variance (5.2)–(5.3), The third column is noise level estimate given by (5.12). A Gaussian noise was added with $\beta = 0.4$ for the first image and $\beta = 0.12$ for the second and third images.

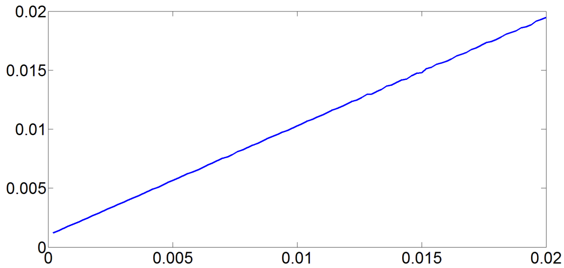


Fig. 6: Noise level estimate as a function of the ground-truth added noise level for image *Barbara*.

Let's go back to the previously mentioned problem of underestimation of the noise level along the edges. Presumably this effect happens because real edges do not match the model of the ideal step edge (5.1). To illustrate this, a set of circles with a variable blur is presented in fig. 7. It can be seen that the described effect does not exist on sharp step edges, but as the edge blurs this effect appears and intensifies. Thus, in the future it will be necessary to develop the proposed method to improve the edge model (5.1) and the correcting formula (5.12). Presumably this approach will have to include estimation of the edge width (or scale [14]). This hypothesis also partially explains reduction of underestimation when the noise level increases: high noise level leads to perception that edges are blurred; the noise conceals blurriness.

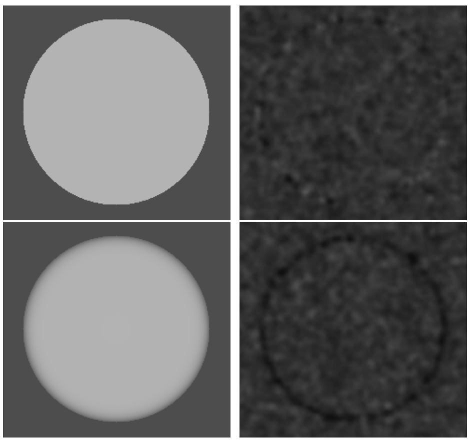


Fig. 7: Noise estimation results on the ideal (top) and non-ideal (bottom) edge.

In fig. 8, filtering results are presented for image *Lena* with three different algorithms: NLM, LJNLM-LR and the proposed HeNLM-LA algorithm. The window size was set to 21×21, the patch size for the NLM algorithm was set to 7×7. The maximal function order for neighborhood expansion in LJNLM-LR and HeNLM-LA algorithms was set to 4. The obtained noise level map was additionally smoothed by the Gaussian filter with $\sigma_2 = 4\sigma$ because the resulting $\beta(x, y)$ is too grainy with a correlation radius on the order of σ, as a result of reduction of high-frequency components in initially uncorrelated (white) noise. The source image was corrupted with a Gaussian noise with $\beta = 0.045$. It can be seen that LJNLM-LR and HeNLM-LA algorithms better reduce noise in fine-textured areas (feathers in the hat, eyes). The NLM algorithm better reduces noise in flat areas (cheeks). The PSNR between the source noise-free and the filtered image is better for the proposed method: 37.5 against 37.44 and 36.84 dB for LJNLM-LR and NLM methods respectively.

Fig. 8: From left to right and top to bottom: the noisy image and the filtering results with NLM, LJNLM-LR and HeNLM-LA.

Fig. 9: Result of the proposed algorithm with irregular noise: noisy image, estimated noise level map, image filtered with a high parameter, with a low parameter, with a locally adaptive parameter.

In fig. 9 the result of a locally adaptive algorithm is shown in comparison with a fixed-parameter filtering. Here the noise level significantly differs for various image fragments. The source image was corrupted by a Gaussian noise with $\beta_1 = 0.045$ for bright areas and $\beta_2 = 0.14$ for dark areas. In fig. 9b the obtained noise estimate is shown. High constant parameter $\rho = 2.5 \cdot \beta_2$ leads to blurring of tile edges in upper left area of the image (fig. 9c), while low parameter $\rho = 2.5 \cdot \beta_1$ leads to insufficient noise removal in the shadow area (fig. 9d). Our adaptive choice of the parameter $\rho = 2.5 \cdot \hat{\beta}(x, y)$ leads to a satisfactory quality of noise removal across the whole image.

7. ACKNOWLEDGMENTS

The research was supported by the RFBR grant 13-07-00584.

СПИСОК ЛИТЕРАТУРЫ

1. *A. Buades.* A non-local algorithm for image denoising // IEEE Conference on Computer Vision and Pattern Recognition, 2005, vol. 2, pp. 60–65

2. *A. Manzanera.* Local Jet based similarity for NL-Means filtering // 20th International Conference on Computer Vision and Pattern Recognition (ICPR), 2010, pp. 2668–2671

3. *S. Wang, Y. Xia, Q. Liu, J. Luo, Y. Zhu, D. Feng.* Gabor feature based nonlocal means filter for textured image denoising // Journal of Visual Communication and Image Representation, vol. 23, issue 7, Oct. 2012, pp. 1008–1018

4. *Yaroslavsky L.P., Kim V.* Rank Algorithms for Picture Processing, Computer Vision // Graphics and Image Processing, vol. 35, 1986, pp. 234–258

5. *Tomasi C., Manduchi R.* Bilateral Filtering for Gray and Color Images // Sixth International Conference on Computer Vision (ICCV'98), 1998, pp. 839–846

6. *W. Forstner.* Image preprocessing for feature extraction in digital intensity, color and range images // Springer Lecture Notes on Earth Sciences, vol. 95, 2000, pp. 165–189

7. *Ce Liu, Szeliski, R., Sing Bing Kang, Zitnick, C.L., Freeman, W.T.* Automatic Estimation and Removal of Noise from a Single Image // IEEE Transactions on Pattern Analysis and Machine Intelligence, Feb. 2008, vol. 30, no. 2, pp. 299–314

8. *St. Pyatykh, J. Hesser, and Lei Zheng.* Image Noise Level Estimation by Principal Component Analysis // IEEE Transactions on Image Processing, 2013, vol. 22, no. 2, pp. 687–699

9. *M.V. Storozhilova, A.S. Lukin, D.V. Yurin, V.E. Sinitsyn.* 2.5D Extension of Neighborhood Filters for Noise Reduction in 3D Medical CT Images // Lecture Notes in Computer Science, 2013, vol. 7870, pp. 1–16

10. *T. Lindeberg.* Scale–Space Theory in Computer Vision // Kluwer Academic Publishers, Dordrecht, 1994

11. *M. Abramowitz, I.A. Stegun.* Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables // U.S. National Bureau of Standards (1964); Dover, New York 1965

12. *A.S. Krylov, A.V. Kutovoi, Wee Kheng Leow.* Texture Parameterization With Hermite Functions // Proc. of The 12th International Conference Graphicon'2002. Russia, Nizhny Novgorod, 2002, pp. 190–194

13. *J. Canny.* A computational approach to edge detection // IEEE PAMI, 1986, vol. 8, pp. 34–43

14. *T. Lindeberg.* Edge Detection and Ridge Detection with Automatic Scale Selection // International Journal of Computer Vision, 1998, vol. 30, no. 2, pp. 117–154