A Noise Removal in Color Images

Thanh N.H. Dang [1], Sergey D. Dvoenko [2]

1) Tula State University, Tula, Russian Federation, dnhthanh@hueic.edu.vn

2) Tula State University, Tula, Russian Federation, dsd@tsu.tula.ru

Abstract: *In this paper, we propose a method to remove noise in RGB-color (Red-Green-Blue) images. This method is based on a total variation of intensity function of images. We propose to remove a linear combination of Gaussian and Poisson noises. This type of noise can be used to well approximate real noises in raster images.*

***Keywords*:** mixed Poisson-Gaussian noise, total variation, Euler-Lagrange equation, noise removal, denoising.

1. INTRODUCTION

In image processing, input image quality usually affects to performance of tasks of image processing. An image quality can be considerably reduced by noise.

The noise removal problem has attracted a lot of attention in recent years. Many researches denote that real noises can be usually well approximated by mixed Poisson-Gaussian noise [1-2].

In practice, the mixed Poisson-Gaussian noise is usually their superposition. This is caused by natural physical process of image formation: Poisson noise is added into image first and Gaussian noise is added later.

In order to remove this superposition of Poisson and Gaussian noises, Luisier et al. proposed the effective and theoretically strong PURE-LET method [3]. However, many parameters need to be evaluated in it. Hence, multiple parameters increase complexity of this method and influence to the final quality of processing.

We suppose that mixed Poisson-Gaussian noise is equivalent to a linear combination of Poisson and Gaussian noises. Hence, we propose a method to remove this linear combination of noises. This method is simpler than the PURE-LET method and includes small number of parameters to be evaluated.

In order to remove noise in grayscale images, we have proposed before a model based on total variation of intensity function of images [4-6]. This model is a linear combination of the ROF [7] model to remove Gaussian noise and the modified ROF [8] model to remove Poisson noise.

We use criteria like *PSNR* (Peak Signal-to-Noise Ratio), *MSE* (Mean Square Error), *SSIM* (Structure SIMilarity) for image quality evaluation after denoising [10-11].

Denoising results compared with different methods for grayscale images have been discussed previously in [4-6] and prove to be very good.

In this paper, we extend the proposed model to remove noise in RGB-color images. Additionally, to remove noise in non RGB-color images, we can transform them into RGB-model [9]. This transformation can be done exactly without any errors or with very small ones. Hence, in this paper, we discuss RGB-color images only.

In experiments, we use real RGB-color images with an artificial mixed noise for each channel separately. This mixed noise is a superposition of Poisson and Gaussian noises generated by the MATLAB built-in function *imnoise*.

However, this function usually changes properties of that mixed noise, because if the intensity value of a pixel after adding noise is out of the interval 0÷255, then the *imnoise* function sets it to 0 or 255.

Nevertheless, in this case we consider values are to be the same with values of corresponding pixels of the original image. Hence, we suppose that this mixed noise is unknown in general.

In order to remove noise in color images, we remove it in every channel R, G and B separately. Then we combine denoising results of all channels to reconstruct the original RGB-color image.

We can do it because intensity values of every channel R, G or B are absolutely independent. Hence, the intensity values of pixels (i,j), $i=1...N_1$, $j=1...N_2$ of a color image are vectors of three components $\boldsymbol{u}_{ij}=(u_{ij}^R,u_{ij}^G,u_{ij}^B)$.

The image quality after denoising is compared with the ROF model, the modified ROF model, and the PURE-LET method only.

Because the denoising process is implemented for every channel separately, we use same notations as in previous works for each color channel. In this paper, we consider our proposed method to remove noise in a single channel (R or G or B).

Comparison of results is also performed for every channel separately by *PSNR*, *MSE*, *SSIM*.

2. NOISE REMOVAL FOR A SINGLE CHANNEL

Let a bounded domain $\Omega\subset\mathbf{R}^2$ be given and functions $u(x,y)\in\mathbf{R}$ and $v(x,y)\in\mathbf{R}$ be, respectively, intensity values of pixels of a single channel of the original and the observed images, where $(x,y)\in\Omega$.

If the function u is smooth, then its total variation is defined by

$$V_T[u]=\int_\Omega|\nabla u|\,dxdy\ ,$$

where $\nabla u=(u_x,u_y)$ is a gradient, $u_x=\partial u/\partial x$, $u_y=\partial u/\partial y$, $|\nabla u|=\sqrt{u_x^2+u_y^2}$. In this paper, we consider that $V_T[u]<\infty$.

Many researches denote that the total variation of an image intensity function characterizes image smoothness. The total variation of a noisy image is always greater than the total variation of the corresponding smooth image [7, 8, 12].

In order to build a model that can remove the combination of Poisson-Gaussian noises, we solve the optimization problem $V_T[u]\to\min$ with a condition: for a given noisy image, we consider the noise variance is unchangeable (Poisson noise is not changed and Gaussian noise is only depends on noise variance):

$$\int_\Omega\ln(p(v\,|\,u))dxdy=const\ , \qquad (1)$$

where $p(v \mid u)$ is a conditional probability of observation of the real image v with given the ideal image u.

The probability density function of Gaussian noise with variance σ^2 is defined by

$$p_1(v \mid u) = \exp\left(-(v-u)^2 / 2\sigma^2\right) \Big/ (\sigma\sqrt{2\pi}),$$

and the discrete probability function of Poisson noise is

$$p_2(v \mid u) = \exp(-u) u^v / v!.$$

In order to combine Gaussian and Poisson noises, we consider the following linear combination

$$\ln(p(v \mid u)) = \lambda_1 \ln(p_1(v \mid u)) + \lambda_2 \ln(p_2(v \mid u)),$$

$$\lambda_1 > 0, \ \lambda_2 > 0, \ \lambda_1 + \lambda_2 = 1.$$

According to (1), we obtain the denoising problem for a single channel:

$$\begin{cases} u^* = \arg\min_u \int_\Omega |\nabla u| \, dxdy \\ \int_\Omega \left(\frac{\lambda_1}{2\sigma^2}(v-u)^2 + \lambda_2 (u - v\ln(u)) \right) dxdy = \kappa, \end{cases}$$

where κ is a constant value.

We rewrite this problem in another form by using Lagrange functional

$$(u^*, \tau^*) = \arg\min_{u,\tau} L(u, \tau), \quad (2)$$

$$L(u,\tau) = \int_\Omega |\nabla u| \, dxdy + \tau\left(\frac{\lambda_1}{2\sigma^2} \int_\Omega (v-u)^2 dxdy + \lambda_2 \int_\Omega (u - v\ln(u)) dxdy - \kappa \right),$$

where $\tau > 0$ is Lagrange multiplier.

3. NUMERICAL SOLUTION OF A MODEL

In this paper, we use the following result to solve the problem (2) [12].

Let a function $f(x,y)$ be defined in a limited domain $\Omega \subset \mathbf{R}^2$ and be second-order continuously differentiated by x and y, where $(x,y) \in \Omega$.

Let $F(x,y,f,f_x,f_y)$ be a convex functional, where $f_x = \partial f / \partial x$, $f_y = \partial f / \partial y$. Then the solution of the following optimization problem

$$\int_\Omega F(x,y,f,f_x,f_y) dxdy \to \min$$

satisfies the following Euler-Lagrange equation

$$F_f(x,y,f,f_x,f_y) - \frac{\partial}{\partial x} F_{f_x}(x,y,f,f_x,f_y) - \frac{\partial}{\partial y} F_{f_y}(x,y,f,f_x,f_y) = 0,$$

where $F_f = \partial F / \partial f$, $F_{f_x} = \partial F / \partial f_x$, $F_{f_y} = \partial F / \partial f_y$.

The solution of the problem (2) satisfies the following Euler-Lagrange equation

$$-\frac{\lambda_1}{\sigma^2}(v-u) + \lambda_2\left(1 - \frac{v}{u}\right) - \mu \frac{\partial}{\partial x}\left(\frac{u_x}{\sqrt{u_x^2 + u_y^2}} \right) - \mu \frac{\partial}{\partial y}\left(\frac{u_y}{\sqrt{u_x^2 + u_y^2}} \right) = 0, \quad (3)$$

where $\mu = 1/\tau$. We rewrite equation (3) in the form

$$\frac{\lambda_1}{\sigma^2}(v-u) - \lambda_2\left(1 - \frac{v}{u}\right) + \mu \frac{u_{xx}u_y^2 - 2u_x u_y u_{xy} + u_x^2 u_{yy}}{(u_x^2 + u_y^2)^{3/2}} = 0, \quad (4)$$

where

$$u_{xx} = \frac{\partial^2 u}{\partial x^2}, \ u_{yy} = \frac{\partial^2 u}{\partial y^2},$$

$$u_{xy} = \frac{\partial}{\partial x}\left(\frac{\partial u}{\partial y}\right) = \frac{\partial}{\partial y}\left(\frac{\partial u}{\partial x}\right) = u_{yx}.$$

In order to obtain the discrete form of the model (4), we use the similar idea that Rudin et al. used [7]. We add an artificial time parameter and consider the function $u = u(x,y,t)$. The equation (4) corresponds to the following diffusion equation

$$u_t = \frac{\partial u}{\partial t} = \frac{\lambda_1}{\sigma^2}(v-u) - \lambda_2\left(1 - \frac{v}{u}\right) + \mu \frac{u_{xx}u_y^2 - 2u_x u_y u_{xy} + u_x^2 u_{yy}}{(u_x^2 + u_y^2)^{3/2}}. \quad (5)$$

Then the discrete form of the equation (5) is

$$u_{ij}^{k+1} = u_{ij}^k + \xi\left(\frac{\lambda_1}{\sigma^2}\left(v_{ij} - u_{ij}^k\right) - \lambda_2\left(1 - \frac{v_{ij}}{u_{ij}^k}\right) + \mu\varphi_{ij}^k \right), \quad (6)$$

$$\varphi_{ij}^k = \frac{\nabla_{xx}(u_{ij}^k)(\nabla_y(u_{ij}^k))^2}{((\nabla_x(u_{ij}^k))^2 + (\nabla_y(u_{ij}^k))^2)^{3/2}} + \frac{-2\nabla_x(u_{ij}^k)\nabla_y(u_{ij}^k)\nabla_{xy}(u_{ij}^k) + (\nabla_x(u_{ij}^k))^2 \nabla_{yy}(u_{ij}^k)}{((\nabla_x(u_{ij}^k))^2 + (\nabla_y(u_{ij}^k))^2)^{3/2}},$$

$$\nabla_x(u_{ij}^k) = \frac{u_{i+1,j}^k - u_{i-1,j}^k}{2\Delta x},$$

$$\nabla_y(u_{ij}^k) = \frac{u_{i,j+1}^k - u_{i,j-1}^k}{2\Delta y}, \ \nabla_{xx}(u_{ij}^k) = \frac{u_{i+1,j}^k - 2u_{ij}^k + u_{i-1,j}^k}{(\Delta x)^2},$$

$$\nabla_{yy}(u_{ij}^k) = \frac{u_{i,j+1}^k - 2u_{ij}^k + u_{i,j-1}^k}{(\Delta y)^2},$$

$$\nabla_{xy}(u_{ij}^k) = \frac{u_{i+1,j+1}^k - u_{i+1,j-1}^k - u_{i-1,j+1}^k + u_{i-1,j-1}^k}{4\Delta x \Delta y},$$

$$u_{0j}^k = u_{1j}^k; \ u_{N_1+1,j}^k = u_{N_1,j}^k; \ u_{i0}^k = u_{i1}^k; \ u_{i,N_2+1}^k = u_{i,N_2}^k;$$

$$i = 1,\dots,N_1; \ j = 1,\dots,N_2;$$

$$k = 0,1,\dots,K; \ \Delta x = \Delta y = 1; \ 0 < \xi < 1,$$ where

K is enough great number, $K = 500$.

4. PARAMETER OPTIMIZATION

The procedure (6) removes noise in image for given values of parameters $\lambda_1, \lambda_2, \mu, \sigma$. To process the real images with unknown noise, we need to define them. Then we have to change $\lambda_1, \lambda_2, \mu$ in procedure (6) into $\lambda_1^k, \lambda_2^k, \mu^k$ on every k-step.

In the new procedure, these parameters are calculated on every iteration step.

Optimal λ₁ and λ₂. If (u, τ) is a solution of the problem (2), then we obtain the following condition

$$\frac{\partial L(u,\tau)}{\partial u} = 0 .$$

We can use this condition to find optimal parameters

$$\lambda_1 = \frac{\int_\Omega \left(1-\frac{v}{u}\right) dxdy}{\frac{1}{\sigma^2}\int_\Omega (v-u)\, dxdy + \int_\Omega \left(1-\frac{v}{u}\right) dxdy}, \quad \lambda_2 = 1-\lambda_1 .$$

Its discrete form is

$$\lambda_1^k = \frac{\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\left(1-\frac{v_{ij}}{u_{ij}^k}\right)}{\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\left(\frac{v_{ij}-u_{ij}^k}{\sigma^2}+1-\frac{v_{ij}}{u_{ij}^k}\right)}, \quad \lambda_2^k = 1-\lambda_1^k ,$$

where $k = 0,1,...,K$.

Optimal μ. In order to find the optimal parameter μ, we multiply (3) by $(v-u)$ and integrate by parts over domain Ω . Finally, we obtain the optimal parameter

$$\mu = \frac{\int_\Omega \left(-\frac{\lambda_1}{\sigma^2}(v-u)^2 - \lambda_2 \frac{(v-u)^2}{u}\right) dxdy}{\int_\Omega \left(\sqrt{u_x^2+u_y^2} - \frac{u_x v_x + u_y v_y}{\sqrt{u_x^2+u_y^2}}\right) dxdy} .$$

Its discrete form is

$$\mu^k = \frac{\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\left(-\frac{\lambda_1^k}{\sigma^2}(v_{ij}-u_{ij}^k)^2 - \lambda_2^k \frac{(v_{ij}-u_{ij}^k)^2}{u_{ij}^k}\right)}{\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\eta_{ij}^k},$$

where

$$\eta_{ij}^k = \sqrt{(\nabla_x(u_{ij}^k))^2 + (\nabla_y(u_{ij}^k))^2} - \frac{\nabla_x(u_{ij}^k)\nabla_x(v_{ij}) + \nabla_y(u_{ij}^k)\nabla_y(v_{ij})}{\sqrt{(\nabla_x(u_{ij}^k))^2 + (\nabla_y(u_{ij}^k))^2}},$$

$$\nabla_x(u_{ij}^k) = \frac{u_{i+1,j}^k - u_{i-1,j}^k}{2\Delta x}, \quad \nabla_y(u_{ij}^k) = \frac{u_{i,j+1}^k - u_{i,j-1}^k}{2\Delta y},$$

$$\nabla_x(v_{ij}^k) = \frac{v_{i+1,j}^k - v_{i-1,j}^k}{2\Delta x} v, \quad \nabla_y(v_{ij}^k) = \frac{v_{i,j+1}^k - v_{i,j-1}^k}{2\Delta y},$$

$$u_{0j}^k = u_{1j}^k;\ u_{N_1+1,j}^k = u_{N_1,j}^k;\ u_{i0}^k = u_{i1}^k;\ u_{i,N_2+1}^k = u_{i,N_2}^k;$$

$$v_{0j} = v_{1j};\ v_{N_1+1,j} = v_{N_1 j};\ v_{i0} = v_{i1};\ v_{i,N_2+1} = v_{i,N_2};$$

$$i = 1,...,N_1;\ j = 1,...,N_2;\ k = 0,1,...,K;\ \Delta x = \Delta y = 1 .$$

Optimal σ. For the parameter σ, we use the result of Immerker [13]:

$$\sigma = \frac{\sqrt{\pi/2}}{6(N_1-2)(N_2-2)}\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}| u_{ij} * \Lambda |, \text{ where}$$

$$\Lambda = \begin{pmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{pmatrix} \text{ is the mask of an image.}$$

Operator * is a convolution, where

$$u_{ij} * \Lambda = u_{i-1,j-1}\Lambda_{33} + u_{i,j-1}\Lambda_{32} + u_{i+1,j-1}\Lambda_{31} + u_{i-1,j}\Lambda_{23} + u_{ij}\Lambda_{22} + u_{i+1,j}\Lambda_{21} + u_{i-1,j+1}\Lambda_{13} + u_{i,j+1}\Lambda_{12} + u_{i+1,j+1}\Lambda_{11},$$

$$i = 1,...,N_1;\ j = 1,...,N_2;$$

$u_{ij} = 0$, if $i = 0$, or $j = 0$, or $i = N_1 + 1$, or $j = N_2 + 1$.

Parameter σ is calculated at the first step of iteration.

5. IMAGE QUALITY EVALUATION

In order to evaluate an image quality after denoising, we use criteria *PSNR*, *MSE* and *SSIM*:

$$Q_{MSE} = \frac{1}{N_1 N_2}\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}(v_{ij}-u_{ij})^2, \quad Q_{PSNR} = 10\lg\left(\frac{L^2}{Q_{MSE}}\right),$$

$$Q_{SSIM} = \frac{(2\bar{u}\,\bar{v} + C_1)(2\sigma_{uv} + C_2)}{(\bar{u}^2 + \bar{v}^2 + C_1)(\sigma_u^2 + \sigma_v^2 + C_2)},$$

where

$$\bar{u} = \frac{1}{N_1 N_2}\sum_{i=1}^{N_1}\sum_{j=1}^{N_2} u_{ij}, \quad \bar{v} = \frac{1}{N_1 N_2}\sum_{i=1}^{N_1}\sum_{j=1}^{N_2} v_{ij} .$$

$$\sigma_u^2 = \frac{1}{N_1 N_2 - 1}\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}(u_{ij}-\bar{u})^2,$$

$$\sigma_v^2 = \frac{1}{N_1 N_2 - 1}\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}(v_{ij}-\bar{v})^2,$$

$$\sigma_{uv} = \frac{1}{N_1 N_2 - 1}\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}(u_{ij}-\bar{u})(v_{ij}-\bar{v}),$$

$$C_1 = (K_1 L)^2,\ C_2 = (K_2 L)^2;\ K_1 << 1;\ K_2 << 1 .$$

For example, $K_1 = K_2 = 10^{-6}$, L is an image intensity with $L = 2^8 - 1 = 255$ for 8-bits grayscale image.

The lower the value of Q_{MSE}, the better the result of restoration. The greater the value of Q_{PSNR}, the better the image quality. The greater the value of Q_{SSIM}, the better the image quality.

These criteria are just applied to the case, if the original image is given. If the original image is not given, we need to use another criterion like *BRISQUE* (we have used it in previous work [6]). In this paper, we use original images and generated noise. Hence, the criterion *BRISQUE* is not used.

6. EXPERIMENTS

In experiments, we use RGB-color images and generate artificial noise to include it into them. We use "kids.tif" and "pears.png" images from the database of MATLAB. We crop images to standard size of 256×256 pixels for the PURE-LET method. In order to make the noisy image, we use the function *imnoise* of MATLAB. We add Poisson noise into the original image first and add the Gaussian noise into the Poisson noisy image.

The function *imnoise* generates noise for every channel of the original color image $\boldsymbol{u} = (u^R, u^G, u^B)$ independently. Hence, the noise in every channel R, G and B does not depend on noise of other channels.

For the "kids.tif" image, the average variance of Poisson noise of channels R, G and B, respectively, 10.9899, 9.3937, 8.1498. For the "pears.png" image the average variance of Poisson noise of channels R, G and B, respectively, 12.9577, 12.4098, 9.6498. The variance of Gaussian noise is two times greater than the average variance of Poisson noise in both cases.

We suppose that noise (superposition of Poisson and Gaussian noises) is unknown real noise. We also suppose this noise is equivalent to a linear combination of Gaussian and Poisson noises with unknown parameters. Hence, we use the proposed method for each color channel with automatically defined parameters.

In order to remove noise in these color images, we divide intensity function of the observed color image $\boldsymbol{v} = (v^R, v^G, v^B)$ into three channels v^R, v^G, v^B. The denoising process is implemented for every channel independently to get denoised versions u^R, u^G, u^B. Finally, we combine values u^R, u^G, u^B for every channel to reconstruct the original color image $\boldsymbol{u}$.

Results of processing of RGB-color images are showed in Table 1 and Fig. 1. Parameter values of proposed method are also showed in Table 1.

In these experiments our results show that for given real RGB-color images with the unknown mixed Poisson-Gaussian noise, our proposed method more effectively removes noise on every channel than other methods, such as the ROF model, the modified ROF model, and the PURE-LET method.

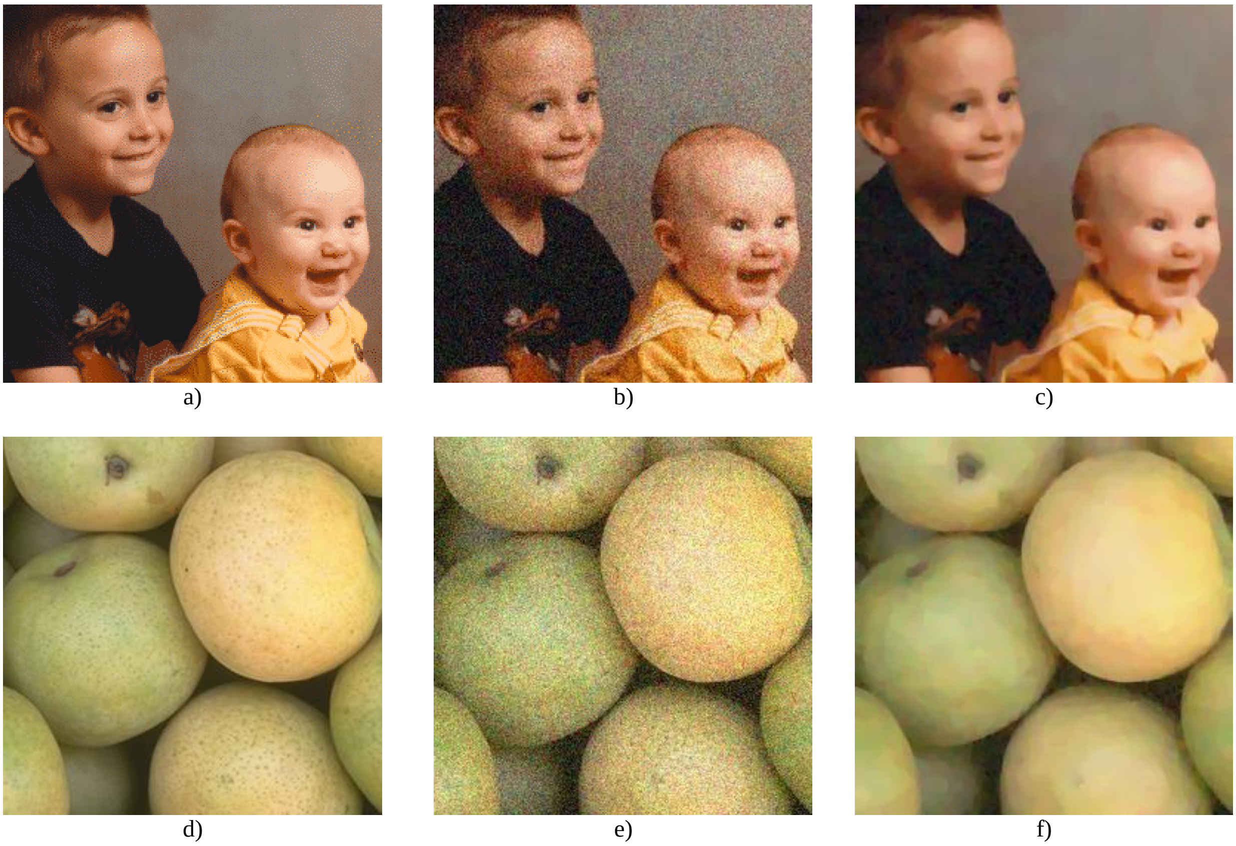

Fig. 1. Noise removal in real images
a) original image "kids.tif", b) noisy image a), c) denoised image b), d) original image "pears.png", e) noisy image d), f) denoised image e)

7. CONCLUSION

In this paper, we proposed a novel method that can effectively remove the real noise in RGB-color images.

In proposed method, we consider the real noise is equivalent to unknown linear combination of Gaussian and Poisson noises. In particular, this method also effectively removes Gaussian noise or Poisson noise separately. This method is based on the total variation of intensity function of images.

The denoising result for every channel (R, G or B) depends on values of coefficients of a linear combination λ_1 and λ_2. These values can be set manually or can be defined automatically. The proposed method with automatically defined parameters is always used to remove the unknown real noise.

The proposed method usually more effective removes the superposition of Poisson and Gaussian noises than the PURE-LET method. Although the PURE-LET method is the special method to remove this noise, many parameters need to be evaluated in it. Such multiple parameters increase complexity of this method and influence to the quality of processing.

Our proposed method includes significantly smaller number of parameters than the PURE-LET method. Hence, we usually get better results than the PURE-LET method.

Our method can be applied to remove noises not only in RGB-images, but also in images with other color models. Then we need to transform that color model to RGB-model and after denoising to original color model again.

8. ACKNOWLEDGMENTS

Our thanks for partial support by the Grant 16-07-01039 of the Russian Foundation for Basic Research.

9. REFERENCES

[1] Alessandro L., Serena M., Fiorella S., You-Wei W. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*. Taylor&Francis 2 (1) (2014). p. 12-24.

[2] Yoan L.M., Elsa D.A., Jean-Christophe O. An Unbiased Risk Estimator for Image Denoising in the Presence of Mixed Poisson–Gaussian Noise, *IEEE transaction on Image processing* 23 (3) (2014). p. 1255-1268.

[3] F. Luisier, T. Blue, M. Unser. Image denoising in mixed Poisson-Gaussian noise, *IEEE transaction on Image Processing* 20 (3) (2011). p. 696-708.

[4] D.N.H. Thanh, S.D. Dvoenko. A variational method to remove the combination of Poisson and Gaussian noises. *Proceeding of the 5-th international workshop on Image mining: Theory and Application*. Berlin, 2015, pp. 38-45.

[5] Thanh D.N.H, Dvoenko S.D. A denoising of biomedical images. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Photogrammetric techniques for video surveillance, biometrics and biomedicine*. Moscow, 2015. V. XL-5/W6. pp. 73-78.

[6] D.N.H. Thanh, S.D. Dvoenko. A method of Total variation to remove the mixed Poisson-Gaussian noise, *Pattern recognition and Image analysis* 26 (2) (2016). p. 285-293.

[7] Rudin L.I., Osher S., Fatemi E. *Nonlinear total variation based noise removal algorithms*, Physica D (60) (1990). p. 259-268.

[8] Le T., Chartrand R., Asaki T.J. A variational approach to reconstructing images corrupted by Poisson noise, *Journal of mathematical imaging and vision* 27 (3) (2007). p. 257-263.

[9] Haim Levkowitz. Color Theory and Modeling for Computer Graphics, Visualization, and Multimedia Applications. *Color Theory and Modeling for Computer Graphics, Visualization, and Multimedia Applications* (402) (1997). p. 45-74.

[10] Wang Z. et al. Image quality assessment: From error visibility to structural similarity, *IEEE transaction on Image processing* 13 (4) (2004). p. 600-612.

[11] Wang Z., Bovik A.C. *Modern image quality assessment*. Morgan & Claypool Publisher, 2006. p. 146.

[12] Zeidler E. Nonlinear functional analysis and its applications: *Variational methods and optimization*. Springer, 1985. p. 662.

[13] Immerker J. Fast noise variance estimation, *Computer vision and image understanding* 64 (2) (1996). p. 300-302.

Table 1. – Quality comparison of noise removal in RGB-color images.

		Image “kid.tif”			Image “pears.png”		
	Channel	**R**	**G**	**B**	**R**	**G**	**B**
Noisy	Q_{PSNR}	21.2978	22.367	23.628	19.9164	20.0031	22.1598
	Q_{MSE}	482.284	377.0333	282.0215	662.8813	649.7824	395.457
	Q_{SSIM}	0.3874	0.3681	0.4581	0.2163	0.2202	0.2903
ROF	Q_{PSNR}	27.0423	28.4412	27.9435	29.8793	30.0688	30.1198
	Q_{MSE}	129.8466	101.6742	105.7622	93.6708	92.5037	90.7653
	Q_{SSIM}	0.6367	0.7215	0.6733	0.6949	0.7011	0.7223
Modified ROF	Q_{PSNR}	26.1688	26.8744	26.6617	28.4441	28.6805	28.877
	Q_{MSE}	137.9981	110.8799	115.8765	135.2315	133.1886	130.7961
	Q_{SSIM}	0.5466	0.6218	0.5633	0.6927	0.6811	0.6734
PURE-LET	Q_{PSNR}	27.0433	29.2241	28.1743	31.0377	31.0522	31.3566
	Q_{MSE}	129.2266	75.7599	99.7746	52.2663	52.4019	48.7322
	Q_{SSIM}	0.637	0.7328	0.6941	0.7812	0.7545	0.7452
Proposed method	Q_{PSNR}	***27.0634***	***29.4706***	***28.2815***	***31.0398***	***31.0727***	***31.4719***
	Q_{MSE}	***127.8611***	***73.4546***	***96.5906***	***51.1801***	***50.794***	***46.3328***
	Q_{SSIM}	***0.6378***	***0.7336***	***0.6945***	***0.7837***	***0.7576***	***0.7464***
	λ_1 λ_2 σ μ	0.7655, 0.2345, 24.5496, 0.2361.	0.7287, 0.2713, 20.8020, 0.2221.	0.7194, 0.2806, 19.7068, 0.2204.	0.7918, 0.2082, 26.0736, 0.2885.	0.7868, 0.2132, 26.0343, 0.2876.	0.7322, 0.2678, 20.5315, 0.2309.