A Fast and Effective Algorithm for a Poisson Denoising Model With Total Variation

Wei Wang and Chuanjiang He

Abstract—In this letter, we present a fast and effective algorithm for solving the Poisson-modified total variation model proposed in [Le et al., “A variational approach to reconstructing images corrupted by Poisson noise,” J. Math. Imag. Vis., vol. 27, no 3, pp. 257–263, Apr. 2007]. The existence and uniqueness of solution for the model are proved by using a different method. A semi-implicit difference scheme is designed to discretize the derived gradient descent flow with a large time step. Different from the original numerical scheme, our scheme is conditional stable with a less stringent condition and can ensure that the numerical solution is strictly positive in image domain. Experimental results show the efficiency and effectiveness of our algorithm.

Index Terms—Gradient descent, Poisson denoising, semi-implicit scheme, total variation.

I. INTRODUCTION

POISSON noise, also known as photon noise, is a basic form of uncertainty associated with the measurement of light. An image sensor measures scene irradiance by counting the number of photons incident on the sensor over a given time interval. The photon counting is a classic Poisson process that follows Poisson distribution [1]. Poisson noise removal is a fundamental task for many imaging applications where images are generated by photon-counting devices, such as computed tomography, magnetic resonance imaging, and astronomical imaging. Many methods and algorithms have been proposed for Poisson denoising [2]–[21], [24]–[26]. In this letter, we focus on solving the variational Poisson denoising model proposed in [9].

In [9], along the lines of the famous ROF model [22], Le, Chartrand, and Asaki proposed the following Poisson denoising model with total variation regularization (called LCA model in this paper). In detail, if \( f = f(x,y) \) (\( (x,y) \in \Omega \)), a bounded, open subset of \( R^2 \) is an image with Poisson noise, then the denoised image is

\[
 u^* = \arg \inf_u E(u) = \beta \int_\Omega |\nabla u| + \int_\Omega (u - f \log u),
\]

where the constant \( \beta > 0 \) is the regularization parameter, and \( E(u) \) is defined on the set of \( u \in BV(\Omega) \) such that \( \log u \in L^1(\Omega) \); in particular, \( u \) must be positive in \( \Omega \).

The authors used gradient descent with explicit finite difference to solve problem (1). They implemented a straightforward, discretized version of the following PDE:

\[
 \frac{\partial u}{\partial t} = \text{div} \left( \frac{\nabla u}{|\nabla u|} \right) + \beta^{-1} \left( \frac{f}{u} - 1 \right) \quad \text{with} \quad \frac{\partial u}{\partial n} = 0 \quad \text{on} \ \partial \Omega.
\]

Spatial derivatives are computed with standard centered difference approximations. The quantity \( |\nabla u| \) is replaced with \( \sqrt{|\nabla u|^2 + \varepsilon} \) for a small, positive \( \varepsilon \). The time evolution is done with fixed time step \( \tau \), until the change in \( u \) is small enough.

However, the used explicit numerical scheme has two limitations. One problem is that the denoised image \( \tilde{u} \) obtained by this scheme cannot be guaranteed to be positive mathematically. In fact, our experiments show that \( \tilde{u} \) is usually sign changing in \( \Omega \), and so \( E(\tilde{u}) \) is singular due to the presence of \( \log u \) in the functional \( E \). Thus, the sign-changing solution \( \tilde{u} \) is impossibly the best approximation to \( u^* \) in (1). Another problem is that the time step of the used scheme must be small enough to ensure the stability due to the Courant-Friedrichs-Lewy (CFL) condition.

Many algorithms have been proposed for solving the LCA model. In [10], Chan and Chen proposed an efficient multilevel algorithm; however, this algorithm is also fraught with the problem of sign-changing solutions and converges slowly to a stationary solution (see Table I). In [13], Figueiredo and Bioucas-Dias used an alternating direction method of multipliers to solve the LCA model. To address the problem of sign-changing solutions, they replaced \( u \) with the projection \( \max(u,0) \) at each iteration. However, this projection scheme cannot ensure obtaining the positive solution because of \( \max(u,0) = 0 \) for \( u \leq 0 \), thus this algorithm does not necessarily converge to an optimizer of the problem in question.

In this letter, we first prove the existence and uniqueness of solution for the LCA model by a different method. Then a semi-implicit difference scheme is designed to solve numerically the derived gradient descent flow equation of the LCA model. The image restored by this scheme is strictly positive in the image domain, which avoids the problem of sign-changing solution. Besides, the proposed scheme is numerically stable for a large time step.

II. PROPOSED ALGORITHM

In this letter, we propose a new algorithm to solve the LCA model [9]:

\[
 \inf_{u \in C(\Omega)} E(u) = \int_\Omega |\nabla u| + \beta^{-1} \int_\Omega (u - f \log u)
\]

1070-9908 © 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
where \( G(\Omega) = \{ u \in BV(\Omega) : u > 0 \text{ in } \Omega \} \) is a subset of \( BV(\Omega) \) and \( f \) is an original image with Poisson noise.

Inspired by the proof [23, Th. 4.1], we give a new proof of the existence and uniqueness of solution for problem (3), which differs from the proof in [9].

**Theorem 1:** If \( f \in G(\Omega) \cap L^\infty(\Omega) \), then problem (1) has exactly one solution.

**Proof:** Let \( M = \sup f \) and \( m = \inf f \). Since \( H(u) = u - f \log u \) is monotone decreasing in \((0, f)\) and monotone increasing in \((f, +\infty)\), we have

\[
\begin{align*}
\int_{\Omega} \min(u, M) - f \log(\min(u, M)) &= \left( \int_{\Omega(u \leq M)} + \int_{\Omega(u > M)} \right) \min(u, M) - f \log(\min(u, M)) \\
&\leq \int_{\Omega} u - f \log u
\end{align*}
\]

and similarly

\[
\begin{align*}
\int_{\Omega} \max(u, m) - f \log(\max(u, m)) &\leq \int_{\Omega} u - f \log u.
\end{align*}
\]

Since

\[
\begin{align*}
\int_{\Omega} |\nabla \min(u, M)| &= \left( \int_{\Omega(u \leq M)} + \int_{\Omega(u > M)} \right) |\nabla \min(u, M)| \\
&= \int_{\Omega(u \leq M)} |\nabla u| \\
&\leq \int_{\Omega} |\nabla u|
\end{align*}
\]

and

\[
\begin{align*}
\int_{\Omega} |\nabla \max(u, m)| &\leq \int_{\Omega} |\nabla u|
\end{align*}
\]

we have

\[
E(\min(u, M)) \leq E(u)
\]

and

\[
E(\max(u, m)) \leq E(u).
\]

Thus, we can assume \( 0 \leq m \leq u \leq M \text{ a.e. in } \Omega. \)

Since \( f \in G(\Omega) \), we have \( \log f \in L^1(\Omega) \) and \( E(u) \geq \beta^{-1} \int_{\Omega} (u - f \log u) \geq \beta^{-1} \int_{\Omega} (f - f \log f) \), which implies that \( E \) is bounded below in \( G(\Omega) \). Let \( \{ u_n \} \) be the minimizing sequence of problem (3) such that

\[
\lim_{n \to \infty} E(u_n) = \inf_{u \in G(\Omega)} E(u) := E_0.
\]

Then, there is an \( N \) such that, for every \( n > N \)

\[
\int_{\Omega} |\nabla u_n| + \beta^{-1} \int_{\Omega} (u_n - f \log u_n) \leq E_0 + 1
\]

which implies

\[
\int_{\Omega} |\nabla u_n| \leq E_0 + 1 - \beta^{-1} \int_{\Omega} (u_n - f \log u_n) \\
\leq E_0 + 1 - \beta^{-1} \int_{\Omega} f(1 - \log f).
\]

Recalling that \( M \leq u_n \leq m \), thus \( \{ u_n \} \) is bounded in Bounded total variation \( BV(\Omega) \). This implies that there exists a \( u^* \in BV(\Omega) \) such that, up to a subsequence, \( u_n \rightharpoonup u^* \) weakly in \( L^2(\Omega) \) and strongly in \( L^1(\Omega) \). Since \( u_n \geq 0 \text{ a.e. in } \Omega \), we have \( u^* \geq 0 \text{ a.e. in } \Omega \). We further have \( u^* > 0 \text{ a.e. in } \Omega \); otherwise, up to a sequence

\[
\lim_{n \to \infty} \int_{\Omega} - f \log(u_n) = \int_{\Omega} - f \log(u^*) = +\infty
\]

which contradicts with the fact that \( \lim_{n \to \infty} E(u_n) = E_0 \). Thus \( u^* \in G(\Omega) \). Thanks to the lower semicontinuity of the total variation and Fatous lemma, we get that \( u^* \) is a solution of problem (3). The uniqueness of the solution is guaranteed by the strict convexity of problem (3). 

The gradient descent flow of problem (3) is

\[
\frac{\partial u}{\partial t} = \text{div} \left( \frac{\nabla u}{|\nabla u|} \right) - \beta^{-1} \left( 1 - \frac{f}{u} \right).
\]

In the numerical implementation, the quantity \( |\nabla u| \) is replaced by \( \sqrt{|\nabla u|^2 + \epsilon} \) to avoid the denominator being zero, as done in [9].
To solve equation (4) numerically, we design the semi-implicit difference scheme as follows:

$$\frac{u^{n+1} - u^n}{\tau} = \text{div} \left( \frac{-\nabla u^n}{\sqrt{|\nabla u^n|^2 + \epsilon}} \right) - \beta^{-1} \left( 1 - \frac{f}{u^{n+1}} \right)$$

(5)

where $\tau > 0$ is the time step. Equation (5) can be rewritten as

$$(u^{n+1})^2 + a_n u^{n+1} + b = 0$$

(6)

where

$$a_n = -u^n - \tau (\text{div} \left( \frac{-\nabla u^n}{\sqrt{|\nabla u^n|^2 + \epsilon}} \right) - \beta^{-1}), \quad b = -\beta^{-1} \tau f.$$

The restored image is obtained by the positive solution of (6):

$$u^{n+1} = -a_n + \sqrt{a_n^2 - 4b} \over 2$$

(7)

The algorithm is summarized as follows.

**Algorithm 1:**

- Set initial value $u^0 = f$, $n = 0$, predefined iteration number $N$ and tolerance tol $> 0$.
- At the $n$th iteration, update $u^{n+1}$ by (7).
- Go back to step 2, until $n = N$ or $|E(u^{n+1}) - E(u^n)| / |E(u^{n+1})| \leq \text{tol}$.

In general, the implicit scheme can choose a large time step $\tau$, while the explicit scheme must choose a small time step to make the numerical scheme stable due to the CFL condition. Our scheme is semi-implicit, thus, it allows a larger time step than the explicit scheme used in [9].

The following result shows the stability of iteration (7).

**Theorem 2:** If $\tau \beta^{-1} f \geq 1/2$, $\beta^{-1} \geq \sqrt{8}$, then $|u^{n+1}| \leq 2t \beta^{-1} f + f$, where $u^n$ is generated by (7), $t = \tau(n + 1)$.

**Proof:** Let $k$ be the norm of the operator $\text{div}$, then

$$\left| \text{div} \left( \frac{-\nabla u^n}{\sqrt{|\nabla u^n|^2 + \epsilon}} \right) \right| \leq k, \quad \left| \frac{-\nabla u^n}{\sqrt{|\nabla u^n|^2 + \epsilon}} \right| \leq k.$$

and $k \leq \sqrt{8}$ (see [27]). Thus, we have

$$a_n \geq -u^n - \tau (\sqrt{8} - \beta^{-1}).$$

Since $f(x) = \frac{1}{2} (-x + \sqrt{x^2 - 4b})$ is monotone decreasing, we have

$$u^{n+1} \leq u^n + \tau (\sqrt{8} - \beta^{-1}) + \sqrt{(u^n + \tau (\sqrt{8} - \beta^{-1}))^2 - 4b}.$$

It is easy to check that $\sqrt{x^2 + y} \leq x + y$ for $x > 0$ and $y \geq 1$. We, thus, have $u^{n+1} \leq u^n + \tau (\sqrt{8} - \beta^{-1}) - 2b$. Since $\beta^{-1} \geq \sqrt{8}$, we have $u^{n+1} \leq u^n + 2\beta^{-1} f$, and so

$$u^{n+1} \leq f + 2\beta^{-1} \tau (n + 1) f = 2t \beta^{-1} f + f$$

which implies that the sequence $\{u^n\}$ grows linearly with the time $t$. Thus, the iteration by (7) is stable according to [28, Definition A.27].

The following result shows that there exists a $\tau_0 > 0$, such that $E(u^n)$ with $\tau \leq \tau_0$ is monotone decreasing with iteration $n$.

**Theorem 3:** Let the sequence $\{u^n\}$ be generated by Algorithm 1. If

$$\tau \leq \min \frac{\epsilon}{8} \int_{\Omega} |\nabla (u^{n+1} - u^n)|^2 \frac{e}{(|\nabla u|^2 + \epsilon)^{3/2}} dx$$

then, we have $E(u^{n+1}) \leq E(u^n)$.

**Proof:** By the Taylor expansion, we have

$$E(u^{n+1}) - E(u^n) = \int_{\Omega} - \nabla (u^{n+1} - u^n), \frac{\nabla u^n}{\sqrt{|\nabla u^n|^2 + \epsilon}} dx$$

$$+ \beta^{-1} (u^{n+1} - u^n)$$

$$+ \beta^{-1} f (u^{n+1} - u^n) + \beta^{-1} f (u^{n+1} - u^n) \frac{1}{u^{n+1}} dx$$

(8)

Since $\int_{\Omega} (u^{n+1} - u^n) \frac{1}{u^{n+1}} dx \geq \frac{\epsilon}{8} \int_{\Omega} |\nabla (u^{n+1} - u^n)|^2 \frac{e}{(|\nabla u|^2 + \epsilon)^{3/2}} dx$ by [29, Lemma 4.2], we have

$$E(u^{n+1}) - E(u^n) \leq -\frac{1}{8\tau} \int_{\Omega} |\nabla (u^{n+1} - u^n)|^2 \frac{e}{(|\nabla u|^2 + \epsilon)^{3/2}}$$

$$+ \int_{\Omega} |\nabla (u^{n+1} - u^n)|^2 \frac{e}{(|\nabla u|^2 + \epsilon)^{3/2}} dx$$

Thus, if $\tau \leq \min \frac{\epsilon}{8} \int_{\Omega} |\nabla (u^{n+1} - u^n)|^2 \frac{e}{(|\nabla u|^2 + \epsilon)^{3/2}} dx$, we have $E(u^{n+1}) \leq E(u^n)$. ■

**Remark 1:** It is easy to derive that if $|\nabla u| \geq (8\epsilon)^{1/3}$, then the right side of (8) is not less than 1 and so the inequality in (8) holds for any $\tau \leq 1$. In practice, we usually can guarantee $|\nabla u| \geq (8\epsilon)^{1/3}$ because $\epsilon > 0$ is set as a very small value; in this study, $\epsilon = 2e - 16$. Thus, we know by Theorem 3 that $\tau \leq 1$ can usually guarantee that $E(u^{n+1}) \leq E(u^n)$. Fig. 4(a) gives
III. EXPERIMENTS

In this section, we present some experimental results to show the performance of our algorithm, in comparison to other relevant algorithms in [9], [10], [13], and [25], in terms of quality and time. Ten noise-free images shown in Fig. 1 (see Table I for image size) are chosen as test images, from which Poisson noisy images were generated by using the MATLAB command “imnoise” with parameter “Poisson.”

The parameters of the five algorithms for all experiments are set as follows: \( \beta = 0.1, \tau = 0.7, N = 100, \) and \( \text{tol} = 3.0e - 5 \) for our algorithm; \( \beta = 0.1, \) time step \( \tau = 0.2, \) and iteration \( n = 30 \) for the algorithm in [9]; \( \alpha = 0.05, \epsilon = 1.0e - 3 \) for the algorithm in [10]; for the algorithm in [13], regularization parameter \( \tau = 0.1, \mu = 60\tau/M \) (\( M \) is the maximal value of Poisson image) and inner and outer iteration numbers are 10 and 8, respectively; for the algorithm in [25], \( \lambda = 0.5, \) maximal iterations \( \text{Maxiter} = 100 \) and \( \text{RD} = 1.0e - 4 \). We try our best to tune the parameters of the four compared algorithms to obtain the highest signal-to-noise ratio (PSNR) and structural similarity index (SSIM) averagely for the ten test images.

To evaluate the five algorithms quantitatively, PSNR and SSIM are used for measuring the similarity between the denoised image and the corresponding noise-free image. The PSNR and SSIM values for the five algorithms are displayed in Table I. Since the code provided by the authors of [10] can only process \( N \times N \) images, some of PSNR and SSIM values for the algorithm [10] cannot be listed in Table I. From Table I, we can see that the average PSNR and SSIM values of our algorithm are almost same as ones of the algorithm [13], but outperform ones of the algorithms in [9], [10], and [25]. Besides, our algorithm has averagely the minimal runtime among the five algorithms.

Due to the page limitation, in Fig. 2 we only show the denoised images by the five algorithms for Lenna image [see Fig. 1(h)]. From Fig. 2, we can see all of the five algorithms have similar visual effects, which is in accordance with the fact that the largest differences of PSNR and SSIM values for the five algorithms are only 0.57 dB and 0.02, respectively [see Table I(h)].

Fig. 3 gives the denoised images by our numerical algorithm with different values of \( \tau (\tau = 1, 2, 3) \) after 100 iterations. From Fig. 3, it can be seen that our algorithm has almost same visual effect for the three values of \( \tau \). This result verifies experimentally that the proposed numerical scheme is conditionally stable with a less stringent condition.

To see how the energy \( E(u^n) \) and PSNR of our scheme with different \( \tau \) vary with iteration \( n \), in Fig. 4, we give the plots of \( E(u^n) \) versus \( n \) and PSNR versus \( n \) for Fig. 1(a).

Fig. 1. Original images.

Fig. 2. Denoised images by the five algorithms. (a) Noisy image. (b) Algorithm [9]. (c) Algorithm [10]. (d) Algorithm [13]. (e) Algorithm [25]. (f) Ours.

Fig. 3. Denoised images by our algorithm with different \( \tau \) after 100 iterations. (a) \( \tau = 1 \). (b) \( \tau = 2 \). (c) \( \tau = 1 \). (c) \( \tau = 1 \).

Fig. 4. Plots of \( E(u^n) \) and PSNR versus iteration \( n \). (a) \( E(u^n) \) versus \( n \). (b) PSNR versus \( n \).

From Fig. 4(a), we can see that \( E(u^n) \) with \( \tau \leq 1 \) is monotone decreasing with iteration \( n \), but \( E(u^n) \) with \( \tau > 1 \) begins to oscillate after about seven iterations. From Fig. 4(b), we can see that PSNR is monotone increasing if \( n \leq n_r \) and monotone decreasing if \( n > n_r \), where \( n_r \) is a positive integer depending on \( \tau \). Also, our scheme with \( \tau = 0.7 \) obtains the highest PSNR value (32.25 dB) after 15 iterations.

IV. CONCLUSION

In this letter, we proved the existence and uniqueness of the LCA model by a new method. The semi-implicit scheme was designed to discretize the gradient descent flow equation. This scheme can guarantee that the restored image is positive in image domain and allows a larger time step. Experiments show that our algorithm can numerically solve the LCA model quickly and effectively.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their helpful comments and suggestions.
REFERENCES

[1] S. W. Hasinoff, “Photon, Poisson Noise,” Computer Vision: A Reference Guide, Boston, MA: Springer US, 2014, pp. 608–610.

[2] B. Zhang, J. M. Fadili, and J.-L. Starck, “Wavelets, ridgelets, and curvelets for Poisson noise removal,” IEEE Trans. Image Process., vol. 17, no. 7, pp. 1093–1108, Jul. 2008.

[3] C.-A. Deledalle, F. Tupin, and L. Denis, “Poisson NL means: Unsupervised non local means for Poisson noise,” in Proc. IEEE 17th Int. Conf. Image Process., Sep. 2010, pp. 801–804.

[4] J. Salmon, “On two parameters for denosing with non-local means,” IEEE Signal Process. Lett., vol. 17, no. 3, pp. 269–272, Mar. 2010.

[5] J. Salmon, C.-A. Deledalle, R. Willett, and Z. Harmany, “Poisson noise reduction with non-local PCA,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process., 2012, pp. 1109–1112.

[6] A. B. Anc and D. A. M. Nelson, “A nonlocal Poisson denoising algorithm based on stochastic distances,” IEEE Signal Process. Lett., vol. 20, no. 11, pp. 1010–1013, Nov. 2013.

[7] M. Mkitalo and A. Foi, “Optimal inversion of the Anscombe transformation in low-count Poisson image denoising,” IEEE Trans. Image Process., vol. 20, no. 1, pp. 99–109, Jan. 2011.

[8] M. Mkitalo and A. Foi, “Optimal inversion of the generalized Anscombe transformation for Poisson-Gaussian noise,” IEEE Trans. Image Process., vol. 22, no. 1, pp. 91–103, Jan. 2013.

[9] T. Le, R. Chartrand, and T. J. Asaki, “A variational approach to reconstructing images corrupted by Poisson noise,” J. Math. Imag. Vis., vol. 27, no. 3, pp. 257–263, Apr. 2007.

[10] R. H. Chan and K. Chen, “Multilevel algorithm for a Poisson noise removal model with total-variation regularization,” Int. J. Comput. Math., vol. 84, no. 8, pp. 1183–1198, 2007.

[11] R. Zanella, P. Boccacci, L. Zanni, and M. Bertero, “Efficient gradient projection methods for edge-preserving removal of Poisson noise,” Inverse Probl., vol. 25, no. 4, Apr. 2009, Art. no. 045010.

[12] S. Setzer, G. Steidl, and T. Teuber, “Deblurring Poissonian images by split Bregman techniques,” J. Vis. Commun. Image Representation, vol. 21, pp. 193–199, 2010.

[13] M. A. T. Figueiredo and J. M. Bioucas-Dias, “Restoration of Poissonian images using alternating direction optimization,” IEEE Trans. Image Process., vol. 19, no. 12, pp. 3133–3145, Dec. 2010.

[14] X. Liu and L. Huang, “Total bounded variation-based Poissonian images recovery by split Bregman iteration,” Math. Methods Appl. Sci., vol. 35, no. 5, pp. 520–529, Mar. 2012.

[15] X. Wang et al., “Iterative reweighted total generalized variation based Poisson noise removal model,” Appl. Math. Comput., vol. 223, pp. 264–277, Oct. 2013.

[16] W. Zhou and Q. Li, “Adaptive total variation regularization based scheme for Poisson noise removal,” Math. Methods Appl. Sci., vol. 36, no. 3, pp. 290–299, Feb. 2013.

[17] R. Chan, H. Yang, and T. Zeng, “A two-stage image segmentation method for blurry images with Poisson or multiplicative gamma noise”, SIAM J. Imag. Sci., vol. 7, no. 1, pp. 98–127, 2014.

[18] L. Jiang, J. Huang, X.-G. Lv, and J. Liu, “Alternating direction method for the high-order total variation-based Poisson noise removal problem,” Numer. Algorithms, vol. 69, no. 3, pp. 495–516, Jul. 2015.

[19] D. Karimi and R. Ward, “A denoising algorithm for projection measurements in cone-beam computed tomography,” Comput. Biol. Med., vol. 69, pp. 71–82, Feb. 2016.

[20] A. Mansouri et al., “An adaptive spatial–spectral total variation approach for Poisson noise removal in hyperspectral images,” Signal Image Video Process., vol. 10, no. 3, pp. 447–454, Mar. 2016.

[21] X. Liu, “Augmented Lagrangian method for total generalized variation based Poissonian image restoration,” Comput. Math. Appl., vol. 71, no. 8, pp. 1694–1705, Apr. 2016.

[22] L. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation noise removal algorithms,” Physica D, vol. 60, pp. 259–268, 1992.

[23] G. Aubert and J. Aujol, “A variational approach to remove multiplicative noise,” SIAM J. Appl. Math., vol. 68, no. 4, pp. 925–946, 2008.

[24] C. Wu, J. Zhang, and X. Tai, “Augmented Lagrangian method for total variation restoration with non-quadratic fidelity,” Inverse Probl. Imag., vol. 1, no. 1, pp. 237–261, 2011.

[25] Y. Wen, R. Chan, and T. Zeng, “Primal-dual algorithms for total variation based image restoration under Poisson noise,” Sci. China-Math., vol. 59, no. 1, pp. 141–160, 2016.

[26] L. Azzari and A. Foi, “Variance stabilization for noisy plus estimate combination in iterative Poisson denoising,” IEEE Signal Process. Lett., vol. 23, no. 8, pp. 1086–1090, Aug. 2016.

[27] A. Chambolle, “An Algorithm for total variation minimization and applications,” J. Math. Imag. Vis., vol. 20, pp. 89–97, 2004.

[28] G. Aubert and P. Kornprobst, Mathematical Problems in Image Processing, 2nd ed. Berlin, Germany: Springer-Verlag, 2009.

[29] A. Beck and M. Teboulle, “Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems,” IEEE Trans. Image Process., vol. 18, no. 11, pp. 2419–2434, Nov. 2009.