

Multilevel Algorithm for a Poisson Noise Removal Model with Total-Variation Regularization

Raymond H. Chan* and Ke Chen†

(Received 1 March 2007; In final form 11 May 2007)

Many commonly used models for the fundamental image processing task of noise removal can deal with Gaussian white noise. However such Gaussian models are not effective to restore images with Poisson noise, which is ubiquitous in certain applications. Recently Le-Chartrand-Asaki derived a new data-fitting term in the variational model for Poisson noise. This paper proposes a multilevel algorithm for efficiently solving this variational model. As expected of a multilevel method, it delivers the same numerical solution many orders of magnitude faster than the standard single-level method of coordinate descent time-marching. Supporting numerical experiments on 2D gray scale images are presented.

Keywords: Image restoration, Poisson noise, regularization, nonlinear solvers, multilevel methods, sharp edges.

AMS subject class: 68U10, 65F10, 65K10.

1 Introduction

Noise removal is a fundamental task in digital image processing. In many commonly used models [10], images with additive Gaussian white noise have been considered and adequately modeled. In imaging applications where images are generated by photon-counting devices such as computed tomography (CT), Magnetic Resonance Imaging (MRI) and astronomical imaging, Poisson noise rather than Gaussian noise is frequently present. Poisson noise is not additive, and it is image pixel-intensity dependent, i.e. bright pixels are statistically more corrupted than dark pixels.

Although the well-known variational ROF model [31] is effective for restoring a noisy image $z = z(x, y)$, $(x, y) \in \Omega$, with Gaussian noise [5, 10, 31, 34] by

$$\min_u J_1(u), \quad J_1(u) = \int_{\Omega} \left[\bar{\alpha} \sqrt{u_x^2 + u_y^2} + (u - z)^2 / 2 \right] dx dy, \quad (1)$$

*Department of Mathematics, Chinese University of Hong Kong, Shatin, Hong Kong SAR, China.
 Email: rchan@math.cuhk.edu.hk. Web: <http://www.math.cuhk.edu.hk/~rchan>

†Department of Mathematical Sciences, University of Liverpool, Peach Street, Liverpool L69 7ZL, UK. Email: k.chen@liverpool.ac.uk. Web: <http://www.liv.ac.uk/~cmchenke> (for correspondence).

this model is less effective to restore $u = u(x, y)$ when z contains Poisson noise as shown in [20, 21].

In the literature, there were a lot of previous work on modeling of and algorithms for the Poisson noise removal; see [3, 11, 14–16, 19, 20, 23, 24] among others. Here we shall consider the particular total variational model

$$\min_u J(u), \quad J(u) = \int_{\Omega} [\alpha \sqrt{u_x^2 + u_y^2} + (u - z) \log u] dx dy, \quad (2)$$

proposed by Le-Chartrand-Asaki [21] recently, which is within the same model framework of [1, 23, 27, 28]. The total variational (TV) regularization used in (2) using rotationally-invariant TV semi-norm is known to be better than the non-rotationally-invariant semi-norm associated with Markov random field models

$$\min_u J_2(u), \quad J_2(u) = \int_{\Omega} [\alpha(|u_x| + |u_y|) + (u - z) \log u] dx dy, \quad (3)$$

as done in [1, 23, 27, 28]; see [9, 10].

As is widely known, although fundamentally more accurate than linear filtering methods, non-linear variational models are computationally expensive to apply. Multigrid (or multilevel or multi-resolution) methods are well known to exhibit optimal performance whenever they converge. Although their standard variants have convergence difficulties for the highly nonlinear and oscillatory coefficients in these models, we wish to overcome these difficulties for the particular class of models under consideration.

Recently for solving the ROF Gaussian noise model [31], we have proposed robust multilevel methods [8, 9]. For solving Markov random field models for Poisson noise (3), multigrid methods with coarse level functionals using residual information via first order condition on the fine level (along similar lines of [4, 26]) have been considered in [27, 28], where computational efficiency has been achieved. However as remarked in [9], optimization based multigrid methods using first order condition require assumption of the differentiability of the underlying objective functional, hence limit the range of models that can be solved. In fact, once differentiability is assumed, efficient multilevel methods exist; see [12, 32] and the references therein. One also notes that to solve Markov random field type models the graph-cut method [13, 30] may offer a competitive multilevel solution. Such a method does not apply to the TV regularization (2).

Our main attention in this paper is focused on developing a fast and optimization-based multilevel algorithm that can solve the TV regularization and does not involve differentiating the non-smooth functional. Thus we attempt to generalize our recent work of [8, 9] from Gaussian noise removal

to Poisson noise removal. We remark that one may develop a different variational model for Poisson noise removal using the Anscombe transformation [2]:

$\mathbf{y} = 2\sqrt{\mathbf{x} + \frac{3}{8}}$, where \mathbf{x} is a Poisson distributed random variable and \mathbf{y} is an approximate normal distributed random variable [22, 25]. However, the variance will be very large after the transformation.

The plan is to review our recently proposed multilevel method [8] for the Gaussian noise removal for (1) in Section 2. Section 3 presents details of implementation of a multilevel method for model (2). Numerical results are reported in Section 4, where the advantages of (2) over (1) and the efficiency of our multilevel method are highlighted.

2 Review of a multilevel method for optimization

We first review the multilevel method proposed in [8] for solving the Gaussian noise removal problem. The method solves the discretized version of the standard TV model (1):

$$\min_{u \in \mathbb{R}^{n \times n}} J_1(u), \quad (4)$$

where $J_1(u) = \alpha \sum_{i,j=1}^{n-1} \sqrt{(u_{i,j} - u_{i,j+1})^2 + (u_{i,j} - u_{i+1,j})^2} + \sum_{i,j=1}^n \frac{(u_{i,j} - z_{i,j})^2}{2}$, with $\alpha = \bar{\alpha}/h$, $h = 1/(n-1)$ and $z \in \mathbb{R}^{n \times n}$. For simplicity, we shall assume $n = 2^L$.

Let the standard coarsening be used giving rise to $L+1$ levels $k = 1$ (finest), $2, \dots, L, L+1$ (coarsest). Denote the dimension of level k by $\tau_k \times \tau_k$ with $\tau_k = n/2^{k-1}$. Consider the minimization of (4) by the coordinate descent method on the finest level 1:

$$\begin{cases} \text{Given } u^{(0)} = (u_{i,j}^{(0)}) = (z_{i,j}) \text{ with } m = 0, \\ \text{Solve } u_{i,j}^{(m)} = \operatorname{argmin}_{u_{i,j} \in \mathbb{R}} J_{\text{loc}}(u_{i,j}) \text{ for } i, j = 1, 2, \dots, n \\ \text{Set } u^{(m+1)} = (u_{i,j}^{(m)}) \text{ and repeat the above step with } m = m + 1 \\ \text{until a prescribed stopping step on } m, \end{cases} \quad (5)$$

where $J_{\text{loc}}(u_{i,j}) = \alpha \left[\sqrt{(u_{i,j} - u_{i+1,j}^{(m)})^2 + (u_{i,j} - u_{i+1,j}^{(m)})^2} + \sqrt{(u_{i,j} - u_{i-1,j}^{(m)})^2 + (u_{i-1,j}^{(m)} - u_{i-1,j+1}^{(m)})^2} + \sqrt{(u_{i,j} - u_{i,j-1}^{(m)})^2 + (u_{i,j-1}^{(m)} - u_{i+1,j-1}^{(m)})^2} \right] + \frac{1}{2}(u_{i,j} - z_{i,j})^2$. For $u_{i,j}$ at the boundary, Neumann's condition is used, see [8]. Note that each subproblem in (5) is only one dimensional.

What one may find surprising is that method (5) converges quickly but unfortunately to some non-stationary solution (i.e. stuck) near the true solution.

It turns out that the presence of local constants in the solution u is responsible for the stuck minimizer—this is related to the hemi-variateness of u , see [9, 29].

To introduce the multilevel algorithm, it is of interest to rewrite (5) in an equivalent form:

$$\left\{ \begin{array}{l} \text{Given } u^{(0)} = (u_{i,j}^{(0)}) = (z_{i,j}) \text{ with } m = 0, \\ \text{Solve } \hat{c} = \operatorname{argmin}_{c \in \mathbb{R}} J_{\text{loc}}(u_{i,j}^{(m)} + c), \quad u_{i,j}^{(m)} = u_{i,j}^{(m)} + \hat{c} \text{ for } i, j = 1, 2, \dots, n \\ \text{Set } u^{(m+1)} = (u_{i,j}^{(m)}) \text{ and repeat the above step with } m = m + 1 \\ \quad \text{until a prescribed stopping step on } m. \end{array} \right.$$

Here each subproblem can be interpreted as finding the best correction constant at the current approximate $u_{i,j}^{(m)}$ on level 1.

Likewise one may consider a 2×2 block of pixels with pixel values denoted by the current approximate \tilde{u} . We propose to look for the best correction constant to update this block so that the underlying merit functional (relating to all four pixels) achieves a local minimum. One sees that this idea operates on level 2. If we repeat the idea with larger blocks, we arrive at levels 3 and 4 with respective 4×4 and 8×8 blocks.

If we write down the above idea in formulae, it may appear complicated but the idea is simple. On level k , set $b = 2^{k-1}$, $m_1 = (i-1)b + 1$, $m_2 = ib$, $\ell_1 = (j-1)b + 1$, $\ell_2 = jb$. Let $c = (c_{i,j})$. Then the $(i,j)^{\text{th}}$ computational block (stencil) involving the single constant $c_{i,j}$ on level k can be depicted in terms of pixels of level 1 as follows

$$\begin{array}{c|ccc|c} \vdots & \vdots & \dots & \vdots & \vdots \\ \hline \tilde{u}_{m_1-1, \ell_2+1} + c_{i-1, j+1} & \tilde{u}_{m_1, \ell_2+1} + c_{i, j+1} & \cdots & \tilde{u}_{m_2, \ell_2+1} + c_{i, j+1} & \tilde{u}_{m_2+1, \ell_2+1} + c_{i+1, j+1} \\ \tilde{u}_{m_1-1, \ell_2} + c_{i-1, j} & \tilde{u}_{m_1, \ell_2} + c_{i, j} & \cdots & \tilde{u}_{m_2, \ell_2} + c_{i, j} & \tilde{u}_{m_2+1, \ell_2} + c_{i+1, j} \\ \vdots & \vdots & \dots & \vdots & \vdots \\ \hline \tilde{u}_{m_1-1, \ell_1} + c_{i-1, j} & \tilde{u}_{m_1, \ell_1} + c_{i, j} & \cdots & \tilde{u}_{m_2, \ell_1} + c_{i, j} & \tilde{u}_{m_2+1, \ell_1} + c_{i+1, j} \\ \tilde{u}_{m_1-1, \ell_1-1} + c_{i-1, j-1} & \tilde{u}_{m_1, \ell_1-1} + c_{i, j-1} & \cdots & \tilde{u}_{m_2, \ell_1-1} + c_{i, j-1} & \tilde{u}_{m_2+1, \ell_1-1} + c_{i+1, j-1} \\ \vdots & \vdots & \dots & \vdots & \vdots \end{array} \tag{6}$$

Clearly there is only one unknown constant $c_{i,j}$ and we shall obtain a one-dimensional subproblem. After some algebraic manipulation [8, 9], we can rewrite $\min_{c_{i,j}} J(\tilde{u} + P_k c_{i,j})$ (with P_k an interpolation operator distributing

$c_{i,j}$ to a $b \times b$ block on level k as illustrated) as $\min_{c_{i,j}} G(c_{i,j})$:

$$\begin{aligned}
G(c_{i,j}) &= \alpha \sum_{\ell=\ell_1}^{\ell_2} \sqrt{(c_{i,j} - h_{m_1-1,\ell})^2 + v_{m_1-1,\ell}^2} + \alpha \sum_{m=m_1}^{m_2-1} \sqrt{(c_{i,j} - v_{m,\ell_2})^2 + h_{m,\ell_2}^2} + \\
&\quad \alpha \sum_{\ell=\ell_1}^{\ell_2-1} \sqrt{(c_{i,j} - h_{m_2,\ell})^2 + v_{m_2,\ell}^2} + \alpha \sum_{m=m_1}^{m_2} \sqrt{(c_{i,j} - v_{m,\ell_1-1})^2 + h_{m,\ell_1-1}^2} + \\
&\quad \alpha \sqrt{2} \sqrt{(c_{i,j} - \bar{v}_{m_2,\ell_2})^2 + \bar{h}_{m_2,\ell_2}^2} + \frac{1}{2} \sum_{m=m_1}^{m_2} \sum_{\ell=\ell_1}^{\ell_2} (c_{m,\ell} - \tilde{z}_{m,\ell})^2 \\
&= \alpha \sum_{\ell=\ell_1}^{\ell_2} \sqrt{(c_{i,j} - h_{m_1-1,\ell})^2 + v_{m_1-1,\ell}^2} + \\
&\quad \alpha \sum_{m=m_1}^{m_2-1} \sqrt{(c_{i,j} - v_{m,\ell_2})^2 + h_{m,\ell_2}^2} + \\
&\quad \alpha \sum_{\ell=\ell_1}^{\ell_2-1} \sqrt{(c_{i,j} - h_{m_2,\ell})^2 + v_{m_2,\ell}^2} + \alpha \sum_{m=m_1}^{m_2} \sqrt{(c_{i,j} - v_{m,\ell_1-1})^2 + h_{m,\ell_1-1}^2} + \\
&\quad \alpha \sqrt{2} \sqrt{(c_{i,j} - \bar{v}_{m_2,\ell_2})^2 + \bar{h}_{m_2,\ell_2}^2} + \frac{b^2}{2} (c_{i,j} - \tilde{w}_{i,j})^2,
\end{aligned} \tag{7}$$

where we have used the notation:

$$\begin{cases} \tilde{z}_{m,\ell} = z_{m,\ell} - \tilde{u}_{m,\ell}, & \tilde{w}_{i,j} = \text{mean}(\tilde{z}(m_1 : m_2, \ell_1 : \ell_2)) = \frac{1}{b^2} \sum_{m=m_1}^{m_2} \sum_{\ell=\ell_1}^{\ell_2} \tilde{z}(m, \ell), \\ v_{m,\ell} = \tilde{u}_{m,\ell+1} - \tilde{u}_{m,\ell}, & h_{m,\ell} = \tilde{u}_{m+1,\ell} - \tilde{u}_{m,\ell}, \\ \bar{v}_{m_2,\ell_2} = \frac{v_{m_2,\ell_2} + h_{m_2,\ell_2}}{2}, & \bar{h}_{m_2,\ell_2} = \frac{v_{m_2,\ell_2} - h_{m_2,\ell_2}}{2}. \end{cases} \tag{8}$$

The solution of the above local minimization defines $c_{i,j}$.

The proposed piecewise constants based multilevel method for solving (4) will repeatedly update (starting from $\tilde{u} = z$ initially) $\tilde{u} = \tilde{u} + P_k c_{i,j}$, across all levels k . Refer to [7–9].

3 The Poisson noise model and a multilevel method

A commonly-used technique to model Poisson noise is to use an expectation-maximum (EM) algorithm [3, 11, 14–20, 23, 24, 33]. Let $u_{i,j} = u(x_i, y_j)$ be the true image gray level at position (i, j) . Then the observation noisy value $z_{i,j}$

of Poisson distribution with mean $u_{i,j}$ obeys the conditional probability

$$P(f|u) = \prod_{i,j=1}^n P(f_{i,j}|u_{i,j}) = \prod_{i,j=1}^n \frac{e^{-u_{i,j}} u_{i,j}^{z_{i,j}}}{z_{i,j}!}.$$

Then an EM algorithm to solve $\max_u P(f|u)$ is equivalent to

$$\min_u \sum_{i,j=1}^n z_{i,j}! - \log P(f|u) = \sum_{i,j=1}^n (u_{i,j} - z_{i,j} \log u_{i,j}).$$

The continuous formulation for this data-fitting step is the following

$$\min_u \int_{\Omega} (u - z \log u) dx dy,$$

for $u = u(x, y)$. Direct reconstruction solution of this problem does not give good quality images as with all inverse problems. Combined with the TV regularization, the following was proposed in [21]

$$\min_u \int_{\Omega} [\alpha |\nabla u| + (u - z \log u)] dx dy, \quad (9)$$

where it is solved by applying a time-marching method to its Euler-Lagrange equation:

$$\frac{\partial u}{\partial t} = \nabla \cdot \frac{\nabla u}{|\nabla u|} + \frac{1}{\alpha u} (z - u), \quad \frac{\partial u}{\partial \vec{n}} \Big|_{\partial \Omega} = 0. \quad (10)$$

Our task is to solve (9), instead of (10), using a multilevel method. For this purpose, we consider the solution of a discretized form of (9):

$$\min_{u \in \mathbb{R}^{n \times n}} J(u), \quad (11)$$

where $J(u) = \alpha \sum_{i,j=1}^{n-1} \sqrt{(u_{i,j} - u_{i,j+1})^2 + (u_{i,j} - u_{i+1,j})^2} + \sum_{i,j=1}^n (u_{i,j} - z_{i,j} \log u_{i,j})$. As before, level k has $\tau_k \times \tau_k$ blocks with each block having $b_k = 2^{k-1}$ pixels. We should remark that in all of the above equations for the Poisson model we need to impose the constraint $u = \max(u, v_0)$ for some small quantity v_0 (for simplicity assume $v_0 = 10^{-20}$).

3.1 Coordinate descent method

For problem (11), the coordinate descent method defines the iterations:

$$\begin{cases} \text{Given } u^{(0)} = (u_{i,j}^{(0)}) = (z_{i,j}) \text{ with } m = 0, \\ \text{Solve } u_{i,j}^{(m)} = \operatorname{argmin}_{u_{i,j} \in \mathbb{R}} J_{\text{loc}}^1(u_{i,j}) \text{ for } i, j = 1, 2, \dots, n \\ \text{Set } u^{(m+1)} = (u_{i,j}^{(m)}) \text{ and repeat the above step with } m = m + 1 \\ \text{until a prescribed stopping step on } m, \end{cases} \quad (12)$$

on the finest level 1, where $u_{i,j} > 0$ and

$$\begin{aligned} J_{\text{loc}}^1(u_{i,j}) = \alpha & \left[\sqrt{(u_{i,j} - u_{i+1,j}^{(m)})^2 + (u_{i,j} - u_{i+1,j}^{(m)})^2} \right. \\ & + \sqrt{(u_{i,j} - u_{i-1,j}^{(m)})^2 + (u_{i-1,j}^{(m)} - u_{i-1,j+1}^{(m)})^2} \\ & \left. + \sqrt{(u_{i,j} - u_{i,j-1}^{(m)})^2 + (u_{i,j-1}^{(m)} - u_{i+1,j-1}^{(m)})^2} \right] + (u_{i,j} - z_{i,j} \log u_{i,j}). \end{aligned}$$

As we know, the coordinate descent method alone does not lead to a correctly converged solution [6,9]. Next we consider a multilevel algorithm.

3.2 Coarse level problem and solver

Let \tilde{u} be a current approximate solution. On a general level k , with each block of size $b = 2^{k-1}$, assume $b < n$ (otherwise the coarsest level is reached). Define integers m_1, m_2, ℓ_1, ℓ_2 as in the previous section.

Consider the local minimization on the $(i, j)^{\text{th}}$ block of size $b \times b$:

$$c_{i,j} = \operatorname{argmin}_{c_{i,j} \in \mathbb{R}} J_{\text{loc}}^k(\tilde{u}_{i,j} + P_k c_{i,j}) \quad (13)$$

where P_k is as defined by (7) and

$$\begin{aligned} J_{\text{loc}}^k(u_{m,\ell}) = \alpha & \underbrace{\sum_{m,\ell=1}^n \sqrt{(u_{m,\ell} - u_{m+1,\ell})^2 + (u_{m,\ell} - u_{m,\ell+1})^2}}_{\text{Term 1: to use formula (7) above}} \\ & + \underbrace{\sum_{m,\ell=1}^n (u_{m,\ell} - z_{m,\ell} \log u_{m,\ell})}_{\text{Term 2: to be simplified further below}}. \end{aligned}$$

The local problem (13) involves one dimensional minimization for $c_{i,j}$ —

we hope to simplify this formulation. From (6), we know that for the total variation term (Term 1), variable $c_{i,j}$ is only present along 4 sides of the $(i,j)^{th}$ computational block as in (7) because for interior nodes, the $c_{i,j}$ cancels out. For the data-fitting term (Term 2), all pixels of the (i,j) block will involve $c_{i,j}$:

$$\begin{aligned} & \sum_{m,\ell=1}^n \left[(\tilde{u}_{m,\ell} + P_k c_{i,j}) - z_{m,\ell} \log(\tilde{u}_{m,\ell} + P_k c_{i,j}) \right] \\ &= b^2 c_{i,j} - \underbrace{\sum_{m=m_1}^{m_2} \sum_{\ell=\ell_1}^{\ell_2} z_{m,\ell} \log(\tilde{u}_{m,\ell} + c_{i,j})}_{\text{Part of Term 2 depending on } c_{i,j}} + \sum_{m,\ell=1}^n \tilde{u}_{m,\ell}. \end{aligned}$$

The above are combined to give the following equivalent minimization $\min_{c_{i,j}} H(c_{i,j})$ to (13):

$$\begin{aligned} H(c_{i,j}) = & \alpha \sum_{\ell=\ell_1}^{\ell_2} \sqrt{(c_{i,j} - h_{m_1-1,\ell})^2 + v_{m_1-1,\ell}^2} + \alpha \sum_{m=m_1}^{m_2-1} \sqrt{(c_{i,j} - v_{m,\ell_2})^2 + h_{m,\ell_2}^2} + \\ & \alpha \sum_{\ell=\ell_1}^{\ell_2-1} \sqrt{(c_{i,j} - h_{m_2,\ell})^2 + v_{m_2,\ell}^2} + \alpha \sum_{m=m_1}^{m_2} \sqrt{(c_{i,j} - v_{m,\ell_1-1})^2 + h_{m,\ell_1-1}^2} + \\ & \alpha \sqrt{2} \sqrt{(c_{i,j} - \bar{v}_{m_2,\ell_2})^2 + \bar{h}_{m_2,\ell_2}^2} + b^2 c_{i,j} - \sum_{m=m_1}^{m_2} \sum_{\ell=\ell_1}^{\ell_2} z_{m,\ell} \log(\tilde{u}_{m,\ell} + c_{i,j}), \end{aligned} \quad (14)$$

where $v_{m,\ell}$, $h_{m,\ell}$, \bar{v}_{m_2,ℓ_2} , and \bar{h}_{m_2,ℓ_2} are defined in (8).

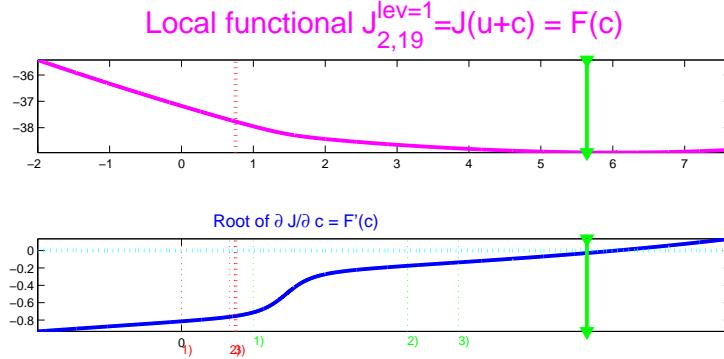
To further put (14) into a more compact form, define $q_1 = 2(\ell_2 + m_2 - m_1 - \ell_1) + 3 = 4b - 1$, $q_2 = (\ell_2 - \ell_1 + 1)(m_2 - m_1 + 1) = b^2$,

$$\alpha_m = \alpha, \quad m = 1, \dots, q_1 - 1; \quad \alpha_{q_1} = \alpha\sqrt{2}.$$

Then pack all quantities $v_{m_1-1,\ell}$, h_{m,ℓ_2} , $v_{m_2,\ell}$, h_{m,ℓ_1-1} , \bar{h}_{m_2,ℓ_2} and $h_{m_1-1,\ell}$, v_{m,ℓ_2} , $h_{m_2,\ell}$, v_{m,ℓ_1-1} , \bar{v}_{m_2,ℓ_2} in (14) respectively into two vectors $\mathbf{a} = (a_m)$ and $\mathbf{b} = (b_m)$. Finally pack elements $z_{m,\ell}$, $\tilde{u}_{m,\ell}$ in block (i,j) respectively into vectors $\bar{z} = (\bar{z}_m)$, $\bar{u} = (\bar{u}_m)$. We can represent (14) as

$$H(c_{i,j}) = \sum_{m=1}^{q_1} \alpha_m \sqrt{a_m^2 + (c_{i,j} - b_m)^2} + b^2 c_{i,j} - \sum_{m=1}^{q_2} \bar{z}_m \log(\bar{u}_m + c_{i,j}) \quad (15)$$

Figure 1. Illustration of the effectiveness of the local smoothers: (17) in vertical dotted lines (inaccurate) and (18) in vertical solid lines (accurate). Here the top plot shows the curve of $F(c) = J(\tilde{u} + ce_{2,19})$ and the bottom of $F'(c)$. The small scripts on the x-axis show the locations of the iterates (the first line of 1), 2), 3) is for method (18) and the second line for method (17).



To solve (15), notice that its first order condition is

$$\frac{\partial H}{\partial c_{i,j}} = \sum_{m=1}^{q_1} \frac{(c_{i,j} - b_m)\alpha_m}{\sqrt{a_m^2 + (c_{i,j} - b_m)^2}} + b^2 - \sum_{m=1}^{q_2} \frac{\bar{z}_m}{(\bar{u}_m + c_{i,j})} = 0. \quad (16)$$

As the second sum in (16) is a high degree polynomial for $c_{i,j}$, one may naturally propose a simple Richardson type smoother

$$\sum_{m=1}^{q_1} \frac{(c_{i,j}^{(\ell+1)} - b_m)\alpha_m}{\sqrt{a_m^2 + (c_{i,j}^{(\ell)} - b_m)^2}} + b^2 - \sum_{m=1}^{q_2} \frac{\bar{z}_m}{(\bar{u}_m + c_{i,j}^{(\ell)})} = 0, \quad \ell = 0, 1, 2, \dots \quad (17)$$

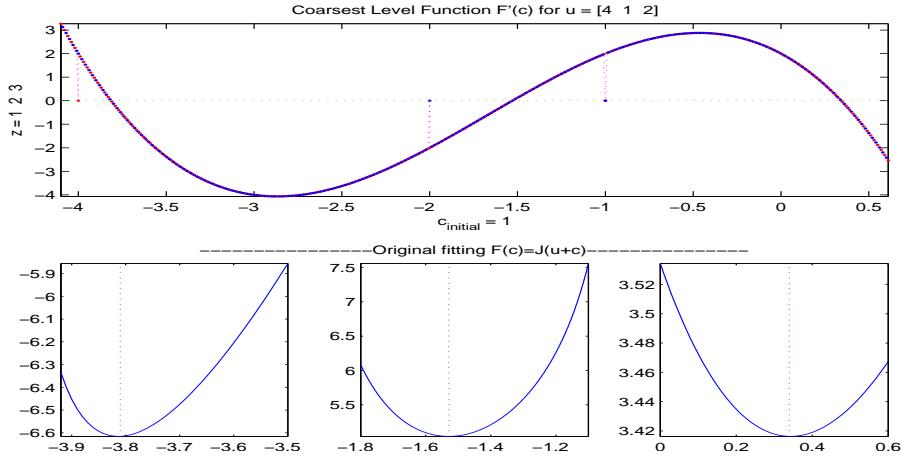
This smoother may converge unfortunately to the non-stationary solution, as shown in Figure 1 where the vertical dotted line shows the converged solution (wrong) from using (17).

What we find useful is the following Richardson type smoother

$$\sum_{m=1}^{q_1} \frac{(c_{i,j}^{(\ell+1)} - b_m)\alpha_m}{\sqrt{a_m^2 + (c_{i,j}^{(\ell)} - b_m)^2}} + \sum_{m=1}^{q_2} \frac{c_{i,j}^{(\ell+1)} + \bar{u}_m - \bar{z}_m}{(\bar{u}_m + c_{i,j}^{(\ell)})} = 0, \quad \ell = 0, 1, 2, \dots \quad (18)$$

In Figure 1, we mark the converged location in solid lines.

Figure 2. Illustration of the non-unique solutions on the coarsest level when constraints are not used. Here the bottom plot shows the curve of $F(c) = J(\tilde{u} + c\mathbf{1})$ with $\mathbf{1}$ a vector of ones and the top of $F'(c)$.



3.3 Coarsest level solver

On the coarsest level, we look for the best constant c that can be added to the current solution \tilde{u} . Hence we solve

$$\min_{c \in \mathbb{R}} \alpha TV(\tilde{u} + c) + \sum_{i,j} [u_{i,j} + c - z_{i,j} \log(\tilde{u}_{i,j} + c)] \quad s.t. \quad \tilde{u}_{i,j} + c > 0$$

which, because of the Neumann boundary condition, is the same as

$$\min_{c \in \mathbb{R}} \sum_{i,j} [u_{i,j} + c - z_{i,j} \log(\tilde{u}_{i,j} + c)] = n^2 c - \sum_{i,j} z_{i,j} \log(\tilde{u}_{i,j} + c) \quad s.t. \quad \tilde{u}_{i,j} + c > 0.$$

Different from the case of the Gaussian denoising [8], the solution to the above problem from solving

$$\sum_{i,j} \frac{z_{i,j}}{\tilde{u}_{i,j} + c} - n^2 = 0$$

is not unique due to multiple roots of the underlying polynomial in c , see Figure 2. Fortunately the constraints $\tilde{u}_{i,j} + c > 0$ can be used to work out an unique solution. In our implementation, we find such an unique solution from using Newton solver with the particular choice $c^{(0)} = \max(z_{i,j} - \tilde{u}_{i,j})$.

3.4 Local patch constants

Whenever the TV semi-norm is used, the solution will allow local constants. Such local constants lead to the hemi-variateness of the solution, which may prevent local minimizations reaching the global minimizer [9,29]. Practically if the coarse mesh does not match up with such constants (patches), the solution near such patches may not be correct.

Following [9], we first detect such patches using the current solution u and then design a special coarse mesh to update each particular constant on a patch. The detection is done by comparing the neighboring consecutive pixels (i_ℓ, j_ℓ) of pixel (i, j) i.e.

$$\text{patch} = \{(i_\ell, j_\ell) \mid |u_{i_\ell, j_\ell} - u_{i, j}| < \varepsilon\} \quad \text{for some small } \varepsilon \text{ (usually } \varepsilon = 10^{-3}\text{).}$$

Assume, at pixel (i, j) , the detected set is

$$S = \{(m, \ell) \mid m_1 \leq m \leq m_2, \ell_1 \leq \ell \leq \ell_2\}.$$

Then the local minimization on the S block of size $b_1 \times b_2$ proceeds similarly to (13)

$$c_{i,j} = \operatorname{argmin}_{c_{i,j} \in \mathbb{R}} J_{\text{loc}}^s(\tilde{u}_{i,j} + P_s c_{i,j}) \quad (19)$$

where P_s on set S (of size $b_1 \times b_2$) is defined similarly to P_k on a set of size $b \times b$ before. That is, all formulations for updating u on a regular block $b \times b$ are applicable to this new block with only minor changes, e.g. b^2 will become $b_1 b_2$.

3.5 Multilevel algorithm

The overall multilevel algorithm proceeds as follows:

Algorithm 3.1 Given an image z containing Poisson noise, set up $L + 1$ levels and assume $\tilde{u} = z$ is an initial guess:

- (i) Let $u_0 = \tilde{u}$.
- (ii) Smooth the approximation on the finest level 1, i.e. solve (12) for $i, j = 1, 2, \dots, n$.
- (iii) On coarse levels $k = 2, 3, \dots, L + 1$:
 - compute $\tilde{z} = z - \tilde{u}$
 - Compute the minimizer c of (14)
 - Add the correction, $\tilde{u} = \tilde{u} + P_k c$.

- (iv) On level $k = 1$, find each patch set S and solve for c (19).
Add each correction $\tilde{u} = \tilde{u} + P_s c$ as with Step 3.
- (v) If $\|\tilde{u} - u_0\|_2$ is small enough, exit with $u = \tilde{u}$ or return to Step 1.

Complexity. For $z \in \mathbb{R}^{n \times n}$, let $N = n^2, n = 2^{L_m}, L \leq L_m$. To estimate the complexity of Algorithm 3.1, we need to estimate the number of terms in (18) using $b_k = 2^{k-1}, \tau_k = N/b_k^2$. Firstly the coefficients a_m, b_m in functional $H(c_{i,j})$ on level k require $4b_k$ flops to compute and secondly the flop count for s steps of the Richardson smoother is $6sq_1 + 4q_2 = 6s(4b_k - 1) + 4sb_k^2$. Hence we can estimate the complexity of one cycle of Algorithm 3.1 as follows

$$\sum_{k=1}^{L+1} (4b_k \tau_k + 6s(4b_k - 1)\tau_k + 4sb_k^2 \tau_k) \approx \sum_{k=1}^{L+1} \left(\frac{4N}{2^{k-1}} + \frac{24sN}{2^{k-1}} + 4sN \right)$$

$$\approx 8N + 48sN + 4sN \log N \approx O(N \log(N)),$$

which is expected of a multilevel method.

Convergence of the algorithm can be established using the classical theory of local minimizations (as in [9]), if the local constraint $u > 0$ is satisfied on all fine levels. It remains to be done for the general case.

4 Numerical experiments

In this section we present some numerical experiments of the proposed multi-level (ML) Algorithm 3.1 to demonstrate the followings:

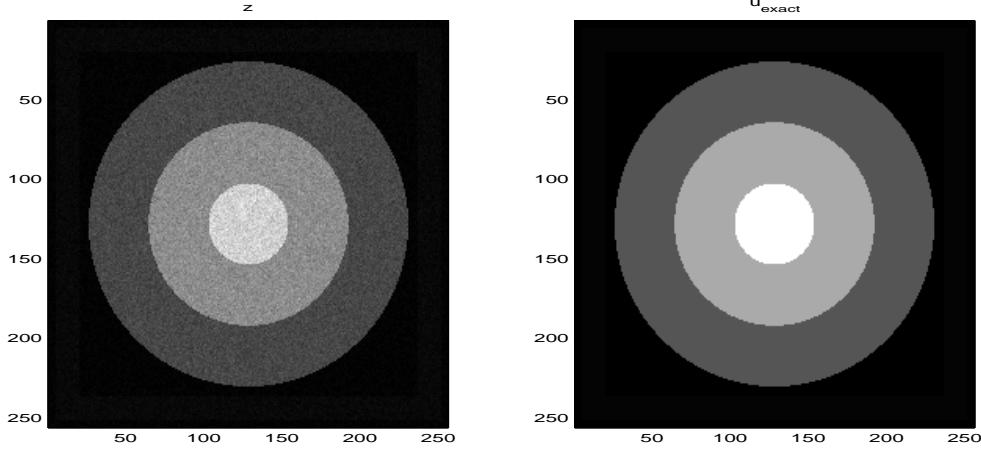
- our new results restore images better than the standard ROF model [31].
- practical performance of our ML algorithm for a range of problems.

Restoration performance is indicated by the mean absolute error (MAE)

$$MAE = MAE(r, u) = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n |r_{i,j} - u_{i,j}|$$

where $r_{i,j}$ and $u_{i,j}$ denote the pixel values of the restored image and the original image respectively, with $u, r \in \mathbb{R}^{m \times n}$. Here we assume $z_{i,j}, r_{i,j}, u_{i,j} \in [0, 255]$.

Firstly we consider the Poisson denoising problem as shown in the left plot of Figure 3 (as considered in [21]). With a fixed parameter $\alpha_{Poisson} = 1/4$ (see (11)), we vary the parameter α for the ROF model in (4) and compare the restoration results of our adopted model and those of the ROF [31] for

Figure 3. A Poisson noise problem (z on the left and the true image u_{exact} on the right).

Gaussian noise. The results are displayed in Figures 4–5, where one observes that, although the ROF results are comparable to the Poisson result in terms of MAE values,

- i) if α is small, the ROF model respects low gray values more and leaves too much noise near large gray level values (see zoomed-in locations on the right of Figure 4 and also the top of the middle plot);
- ii) if α is large, the ROF model tends to lose features at low gray values but recovers well near large gray level values (see zoomed-in locations on the right of Figure 5).

In contrast, the Poisson model treats noise more proportionally with gray level values. This was also observed in [21].

Next we consider four more real-life examples with Poisson noise (see the left plots of Figures 6 and 7). We apply our ML algorithm 3.1 with 4 cycles and display the MAE values of the restored images in Table 1. Clearly ML restored results (using only 4 multilevel cycles) are of good quality.

5 Conclusions

We have generalized an optimization-based multilevel method previously proposed for the standard Gaussian denoising model to solve a Poisson denoising model. The main complication with the non-uniqueness of the coarsest level optimization problem is resolved by making use of the constraints. Our multilevel method has a nearly optimal complexity $O(N \log N)$ and is suitable for quickly processing large images containing Poisson noise.

Figure 4. Performance of ROF [31] with a small $\alpha = 12.5$ for Poisson noise data – data with large noise compromised. Here and also in Figure 5, $MAE(r, z) = 4.96$.

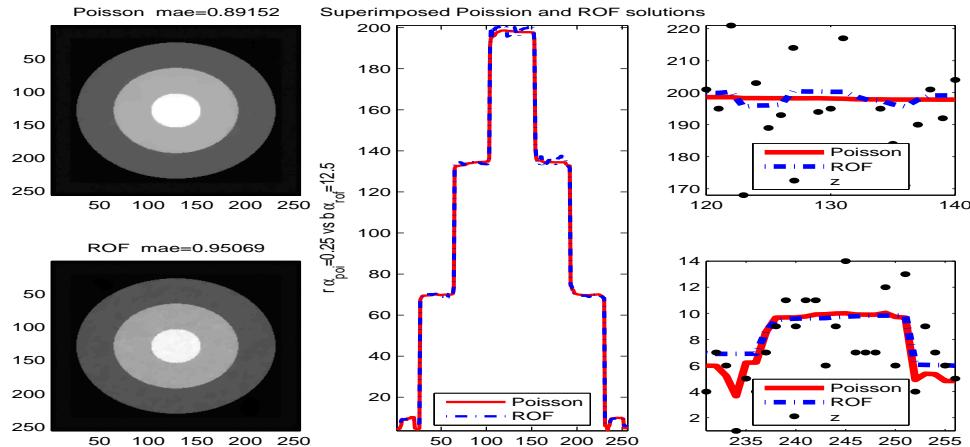
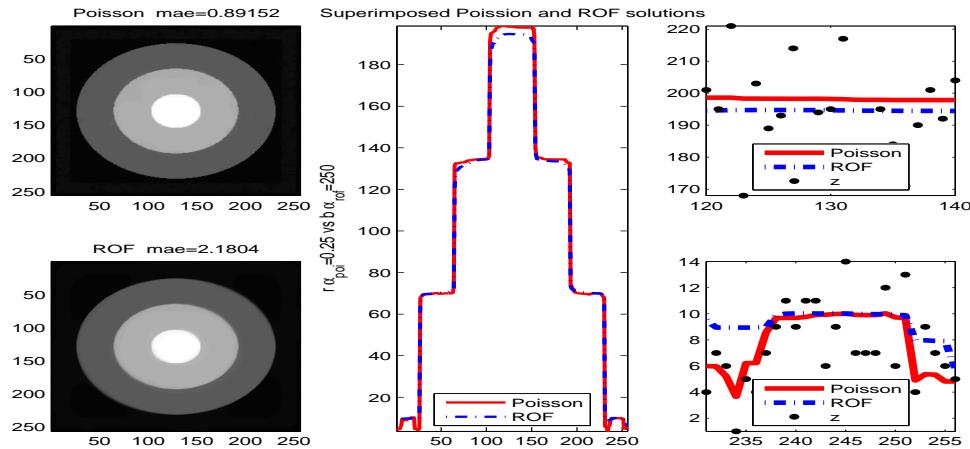


Figure 5. Performance of ROF [31] with a large $\alpha = 250$ for Poisson noise data – data with small noise compromised.



Acknowledgements

The authors wish to thank J. F. Cai and Y. Q. Dong, CUHK, for various discussions and assistance relating to this work. The work of the first author was supported by HKRGC Grant CUHK 400405 and CUHK DAG 2060257. The second author acknowledges the support from the Leverhulme Trust RF/9/RFG/2005/0482 as well as the support and hospitality of the Department of Mathematics, CUHK.

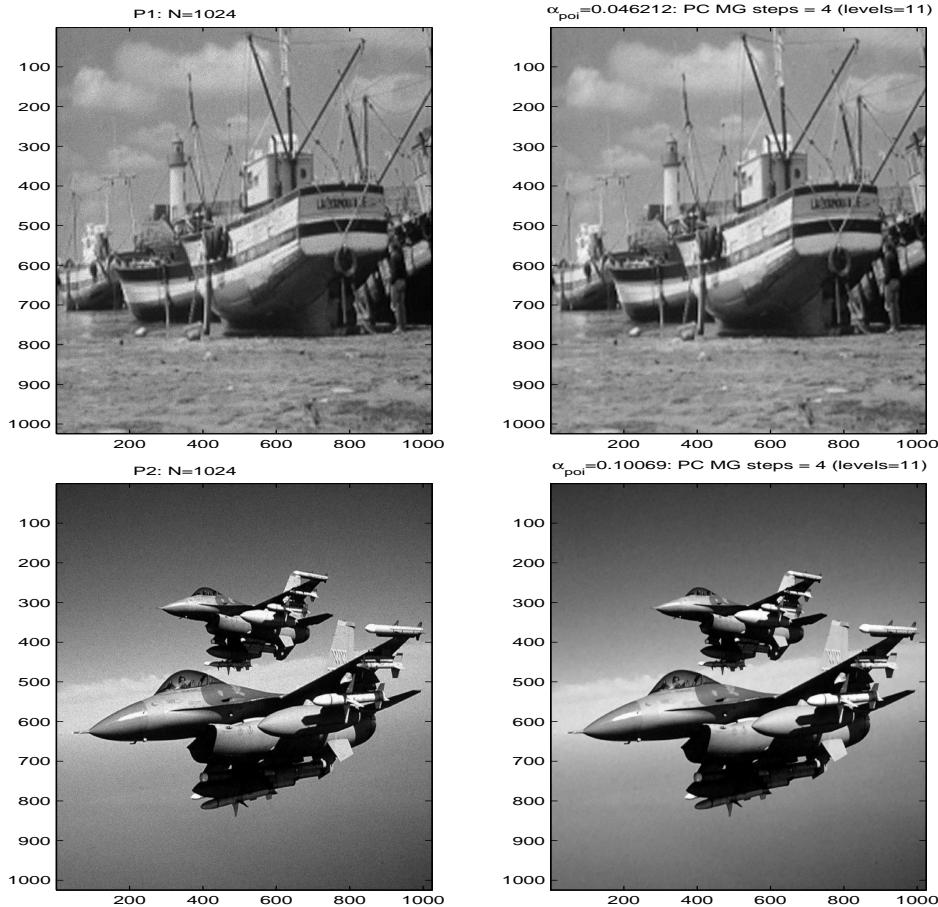
Table 1. Restoration errors of ML with 4 cycles for different problems in varying sizes

Problem	Size n	MAE(r, u_{ML})	Problem	Size n	MAE(r, u_{ML})
1	64	7.05	2	64	4.98
	128	5.62		128	3.60
	256	5.41		256	2.93
	512	3.93		512	2.46
	1024	3.38		1024	2.08
3	64	14.2	4	64	9.98
	128	9.01		128	6.00
	256	4.76		256	3.03
	512	3.53		512	3.70
	1024	3.71		1024	2.25

References

- [1] O. K. Al-Shaykh and R. M. Mersereau, Restoration of lossy compressed noisy images, *IEEE Trans. Image Proc.*, 8 (10), pp. 1348–1360, 1999.
- [2] F. J. Anscombe, The transformation of Poisson, binomial and negative-binomial data, *Biometrika*, 35, pp. 246–254, 1948.
- [3] P. Besbeas, I. D. Fies and T. Sapatinas, A comparative simulation study of wavelet shrinkage estimators for Poisson counts, *Int. Stats Rev.*, 72, pp. 209–237, 2004.
- [4] A. Brandt, Multigrid solvers and multilevel optimization strategies. In J. Cong and J. R. Shinnerl, editors, *Multiscale Optimization and VLSI/CAD*, pp.1-68. Kluwer Academic (Boston), 2000.
- [5] J. F. Cai, R. H. Chan and B. Morini, Minimization of edge-preserving regularization functional by conjugate gradient type methods, *Proc. 1st Int. Conf. PDE-based Image Proc.*, eds. X. C. Tai et al, Springer-Verlag, pp. 109–122, 2006.
- [6] J. L. Carter, *Dual method for total variation-based image restoration*, CAM report 02-13, UCLA, USA, 2002;
see <http://www.math.ucla.edu/applied/cam/index.html>
- [7] R. H. Chan and K. Chen, *Fast multigrid algorithm for a minimization problem in impulse noise removal*, submitted, 2007.
- [8] T. F. Chan and K. Chen, On a nonlinear multigrid algorithm with primal relaxation for the image total variation minimisation, *J. Numer. Algor.*, 41, pp. 387–411, 2006.
- [9] T. F. Chan and K. Chen, An optimization-based multilevel algorithm for total variation image denoising, *SIAM J. Multiscale Modeling and Simulations*, 5, pp. 615–645, 2006.
- [10] T. F. Chan and J. H. Shen, *Image Processing and Analysis - Variational, PDE, wavelet, and stochastic methods*, SIAM Publications, Philadelphia, USA, 2005.
- [11] P. Charbonnier, L. Blanc-Feraud, G. Aubert and M. Barlaud, Deterministic Edge-Preserving Regularization in Computed Imaging, *IEEE Trans. Image Proc.*, 6 (2), pp. 298–311, 1997.
- [12] K. Chen and X. C. Tai, *A nonlinear multigrid method for curvature equations related to total variation minimization*, submitted, 2006.
- [13] J. Darbon and M. Sigelle, Image restoration with discrete constrained total variation Part I: fast and exact optimization, *Journal on Mathematical Imaging and*

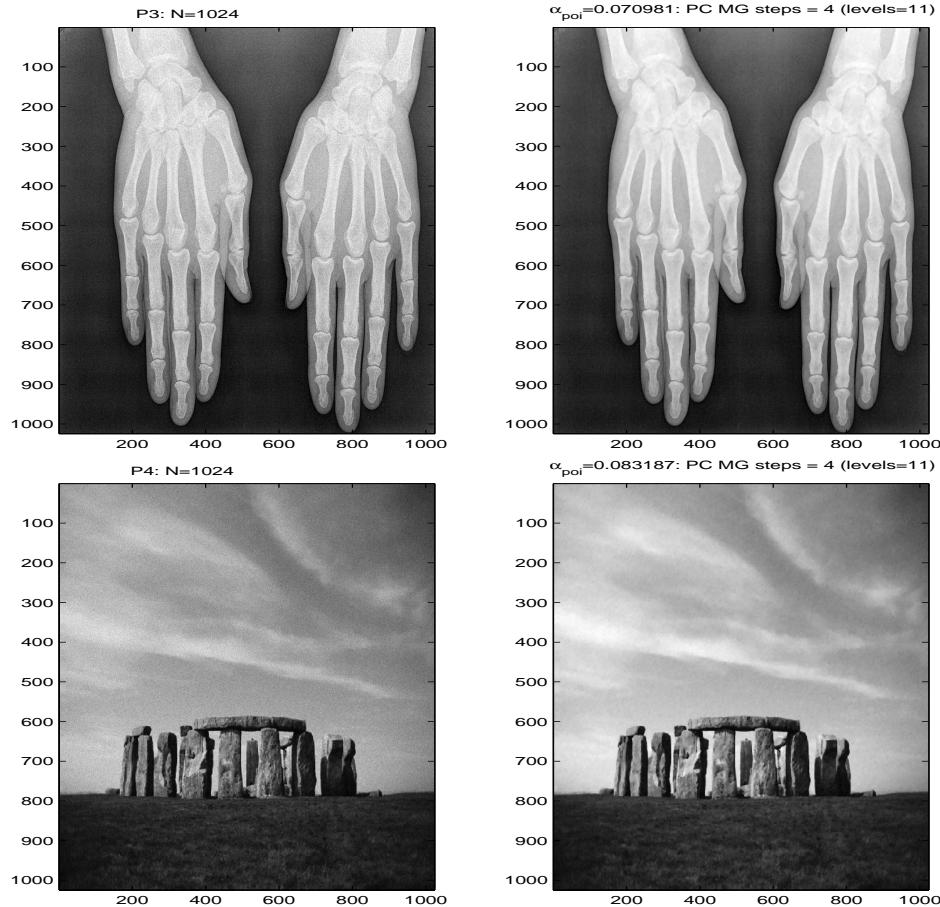
Figure 6. Performance of ML for Problems 1 – 2 (left: Poisson noise vs right: restored image). The restoration quality may be seen from the MAE values: here for Problem 1, $MAE(r, z) = 8.85$, $MAE(r, u) = 3.39$ and for Problem 2, $MAE(r, z) = 7.96$, $MAE(r, u) = 2.08$.



Vision, 26 (3), pp. 261–276, 2006.

- [14] X. H. Han, Y. W. Chen and Z. Nakao, An ICA-based method for Poisson noise reduction, eds. V. Palade, R.J. Howlett, and L.C. Jain, *Lecture Note in Artificial Intelligence* (LNAI) 2773, pp. 1449–1454, 2003, Springer-Verlag.
- [15] E. Josson, S.-C. Huang, and T. F. Chan, *Total variation regularization in positron emission tomography*, UCLA CAM Report 98-48, 1998.
- [16] S. N. Karbelkar, Algorithm for calculating Poisson noise on image intensity correlations, *J. Opt. Soc. Am.*, A/7 (7), 1990.
- [17] C. Kervrann and A. Trubuil, An adaptive window approach for poisson noise reduction and structure preserving in confocal microscopy, in *International Symposium on Biomedical Imaging* (ISBI'04), Arlington, VA, April 2004.
- [18] E. Kolaczyk, Wavelet shrinkage estimation of certain Poisson intensity signals using corrected thresholds, *Statist. Sinica*, 9, pp. 119–135, 1999.
- [19] S. Lasota and W. Niemiro, A version of the Swendsen-Wang algorithm for restora-

Figure 7. Performance of ML for Problems 3 – 4 (left: Poisson noise vs right: restored image). The restoration quality may be seen from the MAE values: here for Problem 3, $MAE(r, z) = 8.16$, $MAE(r, u) = 3.71$ and for Problem 4, $MAE(r, z) = 8.37$, $MAE(r, u) = 2.25$.



tion of images degraded by Poisson noise, *Pattern Recognition*, 36, pp. 931–941, 2003.

- [20] T. Le, R. Chartrand and T. J. Asaki, *Denoising images with Poisson noise statistics*, Los Alamos National Lab, Mathematical Modeling and Analysis Group T-7 summary Report, 2005.
(see <http://math.lanl.gov/Research/Highlights/tvpoisson.shtml>)
- [21] T. Le, R. Chartrand and T. J. Asaki, A variational approach to constructing images corrupted by Poisson noise, UCLA CAM Report 05-49, Sept. 2005. (In print: *J. Math. Imaging and Vision*, 27, 2007.)
- [22] X. Li, H. B. Lu, G. P. Han and Z. R. Liang, A noise reduction method for non-stationary noise model of SPECT sinogram based on Kalman filter, *IEEE Nuclear Science 2001 Symposium Conference Record*, San Diego, 4, pp. 2134–2138, 2001.
- [23] J. Liu and P. Moulin, Complexity-regularized denoising of Poisson-corrupted data, *IEEE Proc. of 2000 Int. Conf. on Image Processing*, 3, pp. 254–257, 2000.

- [24] J. Liu and P. Moulin, Complexity-regularized image denoising, *IEEE Trans. Image Proc.*, 10 (6), pp. 841–851, 2001.
- [25] N. D. A. Mascarenhas, C. A. N. Santos and P. E. Cruijvel, Transmission tomography under Poisson noise using the Anscombe transformation and Wiener filtering of the projections, *Nuclear Instruments and Methods in Physics Research*, A423, pp. 265–271, 1999.
- [26] S. Nash, A multigrid approach to discretized optimization problems, *J. Opt. Methods Softw.*, 14, pp. 99–116, 2000.
- [27] S. Oh, C. A. Bouman and K. J. Webb, Nonlinear multigrid inversion for optical diffusion tomography, *Proc. IEEE Conf. Lasers and Electro-Optics (CLEO)*, 3, pp. 1697–1699, 2005.
- [28] S. Oh, C. A. Bouman and K. J. Webb, Multigrid tomographic inversion with variable resolution data and image spaces, *IEEE Trans. Image Proc.*, 15 (9), pp. 2805–2819, 2006.
- [29] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic press, 1970.
- [30] A. Raj and R. Zabih, A graph cut algorithm for generalized image deconvolution, *Proc. 10th IEEE Int. Conf. on Computer Vision*, 2, pp. 1048–1054, 2005.
- [31] L. I. Rudin, S. Osher and E. Fatemi , Nonlinear total variation based noise removal algorithms, *Physica D*, 60, pp. 259–268, 1992.
- [32] X. C. Tai and J. C. Xu, Global and uniform convergence of subspace correction methods for some convex optimization problems, *Math. Comp.*, 71, pp. 105–124, 2001.
- [33] K. Timmermann and R. Novak, Multiscale modeling and estimation of Poisson processes with applications to photon-limited imaging, *IEEE Trans. Inf. Theor.*, 45, pp. 846–852, 1999.
- [34] C. R. Vogel, *Computational Methods For Inverse Problems*, SIAM publications, USA, 2002.