Variational approach for Gaussian-impulse Noise Removal

Tieyong Zeng
Dept. of Mathematics, HKBU

June 8, 2011
Contributors

- Michael Ng (HKBU)
- Yu Jian (Beijing Jiaotong University)
- Xiao Yu (Beijing Jiaotong University)
Outline

1. Introduction
2. Variational model
3. Algorithm
4. Numerical results
5. Poisson noise removal
6. Discussion
Image denoising

- Gaussian noise
  1. total variation model, wavelet shrinkage
  2. NL-means, K-SVD, BM3D

- Non-Gaussian noise
  1. Impulse noise
  2. Poisson noise
  3. Multiplicative noise
Gaussian noise removal

Noise model

\[ f = u + b, \]

where \( f \) is noisy image, \( u \) is clean image, \( b \) is Gaussian noise.

▶ TV-ROF:

\[ \min_u TV(u) + \lambda \| u - f \|^2, \]

with \( \lambda > 0 \).

▶ discussion

1. advantage: edge preservation
2. disadvantage: texture information lost
K-SVD for Gaussian noise removal

Assumption: Image patches have sparse representation over some hidden dictionary $D$

Variational form

$$\min_{D, \alpha_{i,j}, u} \lambda \| f - u \|^2 + \sum_{i,j \in \mathcal{P}} \| D \alpha_{ij} - R_{i,j} u \|^2 + \sum_{i,j \in \mathcal{P}} \mu_{i,j} \| \alpha_{i,j} \|_0, \quad (1)$$

where $\lambda > 0$, $\mu_{i,j}$ parameters, $R_{i,j}$ extract patches in position $(i,j)$.

Very good results for Gaussian noise removal

Various extensions: inpainting, demosaicking etc

Question: how to extend to other noise?
Impulse noise removal

Noise model

\[ f = \mathbb{N}_{imp}(u), \]

where \( f \) is noisy image, \( u \) is clean image, \( \mathbb{N}_{imp} \) is the impulse procedure.

- \( TV - L_1 \):
  \[
  \min_u TV(u) + \lambda \| u - f \|_1, \]
  with \( \lambda > 0 \).

- Other methods:
  1. tight-frame
  2. AMF filter: detect impulse position first
More challenge case: Impulse-Gaussian noise

Consider mixed noise:

\[ f = \mathbb{N}_{imp}(u + b), \]

where \( \mathbb{N}_{imp} \) is impulse noise, \( b \) is Gaussian noise.

- How to handle?
- Variational approach?
Variational model

In order to restore image from mixed noise:

\[ f = \mathbb{N}_{imp}(u + b), \]

we consider:

- Detect candidature position without impulse noise \( \mathcal{X} \)
- We are then interested in the following denoising model:

\[
\min_{u, D, (\alpha_s)_{s \in \mathcal{P}}} \lambda \left\| \mathcal{X} \otimes (u - f) \right\|_2^2 + \beta \left\| (I - \mathcal{X}) \otimes (u - f) \right\|_1 + \sum_{s \in \mathcal{P}} \left\| (D\alpha_s - R_su) \right\|_2^2 + \sum_{s \in \mathcal{P}} \mu_s \| \alpha_s \|_0,
\]

- Parameters: \( \lambda, \beta, (\mu_s)_{s \in \mathcal{P}} \)
Full algorithm for Gauss-impulse noise removal

- Detect noise free position $\mathcal{X}$ by AMF
- Iteration by Alternating minimization:
  - Given $u$, update each $\alpha_s$ by:
    \[ \hat{\alpha}_s = \arg\min_{\alpha_s} \mu_s \|\alpha_s\|_0 + \|D\alpha_s - R_s u\|_2^2. \] (2)
  - Given $u, (\alpha_s)_{s \in \mathcal{P}}$, update the dictionary by $D$ by:
    \[ \hat{D} = \arg\min_D \sum_{s \in \mathcal{P}} \|D\alpha_s - R_s u\|_2^2. \] (3)
  - Given $D, (\alpha_s)_{s \in \mathcal{P}}$, update $u$ by:
    \[ \hat{u} = \arg\min_u \lambda \|\mathcal{X} \otimes (u - f)\|_2^2 \\
    + \beta \|(I - \mathcal{X}) \otimes (u - f)\|_1 + \sum_{s \in \mathcal{P}} \|D\alpha_s - R_s u\|_2^2, \]
Close form for last step

We denote

\[ W = \sum_{s \in P} R_s^T R_s, \quad M = \sum_{s \in P} R_s^T D_{\alpha_s}, \]

then the last step has close form:

\[ \hat{u}_{ij} = \begin{cases} \frac{M_{i,j} + \lambda f_{i,j}}{W_{i,j} + \lambda}, & \chi_{i,j} = 1 \\ f_{i,j} + \text{shrink} \left( \frac{M_{i,j}}{W_{i,j}} - f_{i,j}, \frac{\beta}{2W_{i,j}} \right), & \chi_{i,j} = 0 \end{cases} \quad (4) \]
### Experimental results

<table>
<thead>
<tr>
<th>Noise</th>
<th>Noisy</th>
<th>AMF</th>
<th>Wang</th>
<th>Mila</th>
<th>Cai1</th>
<th>Cai2</th>
<th>MS</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>15.12</td>
<td>37.24</td>
<td>40.35</td>
<td>43.28</td>
<td>42.86</td>
<td>43.79</td>
<td>46.12</td>
<td><strong>48.11</strong></td>
</tr>
<tr>
<td>20%</td>
<td>12.19</td>
<td>35.07</td>
<td>36.58</td>
<td>39.91</td>
<td>39.33</td>
<td>40.45</td>
<td>42.21</td>
<td><strong>44.05</strong></td>
</tr>
<tr>
<td>30%</td>
<td>10.37</td>
<td>32.71</td>
<td>33.64</td>
<td>37.15</td>
<td>37.00</td>
<td>37.74</td>
<td>38.73</td>
<td><strong>40.43</strong></td>
</tr>
<tr>
<td>40%</td>
<td>9.14</td>
<td>30.98</td>
<td>31.81</td>
<td>35.36</td>
<td>35.38</td>
<td>35.93</td>
<td>36.28</td>
<td><strong>37.92</strong></td>
</tr>
<tr>
<td>50%</td>
<td>8.15</td>
<td>29.22</td>
<td>30.19</td>
<td>33.39</td>
<td>33.73</td>
<td>33.88</td>
<td>33.68</td>
<td><strong>35.27</strong></td>
</tr>
<tr>
<td>60%</td>
<td>7.35</td>
<td>27.43</td>
<td>28.92</td>
<td>31.46</td>
<td>32.21</td>
<td>32.31</td>
<td>31.56</td>
<td><strong>33.06</strong></td>
</tr>
<tr>
<td>70%</td>
<td>6.70</td>
<td>25.75</td>
<td>27.75</td>
<td>29.47</td>
<td>30.53</td>
<td>30.32</td>
<td>29.50</td>
<td><strong>30.75</strong></td>
</tr>
<tr>
<td>80%</td>
<td>6.13</td>
<td>23.88</td>
<td>26.25</td>
<td>27.70</td>
<td>28.29</td>
<td>28.28</td>
<td>27.20</td>
<td><strong>28.40</strong></td>
</tr>
<tr>
<td>90%</td>
<td>5.60</td>
<td>21.04</td>
<td>24.13</td>
<td>24.41</td>
<td>25.13</td>
<td>25.29</td>
<td>25.00</td>
<td><strong>25.70</strong></td>
</tr>
</tbody>
</table>
Comparison

PSNR (dB) for various methods for Barbara with random-valued impulse noise.

<table>
<thead>
<tr>
<th>Noise</th>
<th>Noisy</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ACWMF</td>
</tr>
<tr>
<td>10%</td>
<td>18.84</td>
<td>25.87</td>
</tr>
<tr>
<td>20%</td>
<td>15.83</td>
<td>25.13</td>
</tr>
<tr>
<td>30%</td>
<td>14.05</td>
<td>24.39</td>
</tr>
<tr>
<td>40%</td>
<td>12.82</td>
<td>23.59</td>
</tr>
</tbody>
</table>
**Comparison**

Figure: Denoising for Barbara corrupted by Gaussian noise and salt-and-pepper noise with $\sigma = 10$. From (a) to (f): Noisy image, Cai1, Cai2, MK-SVD, Ours and Clean image.
Comparison

Figure: Denoising results on Lena corrupted by Gaussian noise and salt-and-pepper noise with $\sigma = 5$ and $s = 70\%$: PSNR (dB) values. From (a) to (f): Noisy image, Cai1, Cai2, MK-SVD, Ours and Clean image.
### Comparison

PSNR (dB) for various methods for Lena with Gaussian noise and random-valued impulse noise.

<table>
<thead>
<tr>
<th>σ</th>
<th>r</th>
<th>Noisy</th>
<th>Cai1</th>
<th>Cai2</th>
<th>MK-SVD</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>10%</td>
<td>18.52</td>
<td>32.77</td>
<td>33.78</td>
<td>33.94</td>
<td><strong>34.98</strong></td>
</tr>
<tr>
<td></td>
<td>20%</td>
<td>15.52</td>
<td>31.78</td>
<td>32.37</td>
<td>32.74</td>
<td><strong>33.64</strong></td>
</tr>
<tr>
<td></td>
<td>30%</td>
<td>13.73</td>
<td>30.69</td>
<td>31.21</td>
<td>31.24</td>
<td><strong>32.04</strong></td>
</tr>
<tr>
<td>10</td>
<td>10%</td>
<td>18.19</td>
<td>30.80</td>
<td>31.01</td>
<td>31.60</td>
<td><strong>32.75</strong></td>
</tr>
<tr>
<td></td>
<td>20%</td>
<td>15.38</td>
<td>30.02</td>
<td>30.33</td>
<td>30.80</td>
<td><strong>31.66</strong></td>
</tr>
<tr>
<td></td>
<td>30%</td>
<td>13.67</td>
<td>29.08</td>
<td>29.42</td>
<td>29.69</td>
<td><strong>30.42</strong></td>
</tr>
<tr>
<td>15</td>
<td>10%</td>
<td>17.67</td>
<td>29.13</td>
<td>29.34</td>
<td>29.49</td>
<td><strong>30.85</strong></td>
</tr>
<tr>
<td></td>
<td>20%</td>
<td>15.14</td>
<td>28.59</td>
<td>28.78</td>
<td>28.88</td>
<td><strong>29.98</strong></td>
</tr>
<tr>
<td></td>
<td>30%</td>
<td>13.54</td>
<td>27.96</td>
<td>28.20</td>
<td>28.28</td>
<td><strong>29.11</strong></td>
</tr>
</tbody>
</table>
Learned dictionary

Figure: The learned dictionaries of Barbara under impulse noise with levels 30%, 70% and Gaussian noise with $\sigma = 5$. 
Poisson noise

Mathematically, the probability of observing image $f$ given the true image $u$ is:

$$p(f|u) = \prod_{i,j} e^{-u_{i,j}} u_{i,j}^{f_{i,j}} f_{i,j}!$$

where $f_{i,j}$ denotes the pixel at location $(i,j)$ of the image, and the values of $f$ at every location are independent.
TV-Poisson model

MAP analysis leads to:

\[ \min_u \int_\Omega |\nabla u| + \lambda \int_\Omega (u - f \log u), \]  

(5)

- edge reservation
- texture missing
Variance stabilization transformation (VST)

For Poisson distribution data of mean and variance $\mu$, the VST aims at transforming the data so that the variance is set approximately a constant. Various possibilities:

- Anscombe transform
  \[ \phi_A(x) = 2\sqrt{x + \frac{3}{8}}. \]

- Freeman and Tukey (1950)
  \[ \phi_{FT}(x) = \sqrt{x + 1} + \sqrt{x}. \]

- Square root
  \[ \phi_{SR}(x) = 2\sqrt{x}, \]

- others.
inverse Variance stabilization transformation (iVST)

- Anscombe transform

\[ \varphi_A(y) = \frac{1}{4}y^2 - \frac{3}{8}, \quad y \in \left[ \frac{\sqrt{6}}{2}, +\infty \right) \]

- Freeman and Tukey

\[ \varphi_{FT}(y) = \frac{1}{4}(y - \frac{1}{y})^2, \quad y \in [1, +\infty) \]

- Square root

\[ \varphi_{SR}(y) = \frac{1}{4}y^2, \quad y \in [0, +\infty) \]
Generalized VST

A function pair \((\phi, \varphi)\) where \(\varphi\) is inverse of \(\phi\):
- \(\phi\) is concave, monotone;
- \(\varphi\) is monotone, convex and differentiable.

Challenges:
- Choose \(\phi\) to make the result more Gaussian;
- After \(\phi\), which algorithm to use?
New variational model

Assumption: The patches $R_{\Omega_s}\phi(u), s \in I$ have sparse representation over some dictionary $D$.

- Variational model

$$\min E(u, D, \alpha) := \gamma \int_{\Omega} |\nabla \phi(u)| + \lambda \int_{\Omega} (u - f \log u)$$

$$+ \frac{1}{2} \sum_{s \in I} \|R_{\Omega_s}\phi(u) - D\alpha_s\|^2 + \sum_{s \in I} \mu_s \|\alpha_s\|_0,$$

where $\gamma, \lambda, \mu_s$ are regularization parameters.

- How to solve?
How to solve?

Algorithm:

- learning $D$ from $\phi(f)$ by K-SVD
- find $\alpha_s$ by OMP from $R_s\phi(f)$
- solve:

$$
\min E(u) := \gamma \int_{\Omega} |\nabla \phi(u)| + \lambda \int_{\Omega} (u - f \log u) \\
+ \frac{1}{2} \sum_{s \in I} \| R_{\Omega_s} \phi(u) - D\alpha_s \|^2
$$
Mathematical Properties

Consider:

$$\min E(u) := \gamma \int_{\Omega} |\nabla \phi(u)| + \lambda \int_{\Omega} (u - f \log u)$$

$$+ \frac{1}{2} \sum_{s \in I} \| R_{\Omega_s} \phi(u) - D\alpha_s \|^2$$

- Unique minimizer
- Comparison principle: Suppose that $m_0 = \inf_{s \in \Omega} \inf D\alpha_s \in \mathbb{R}$ and $M_0 = \sup_{s \in \Omega} \sup D\alpha_s \in \mathbb{R}$, then

$$\min(\inf f, \varphi(m_0)) \leq u \leq \max(\sup f, \varphi(M_0)).$$
The last step:

\[
\min \int_{\Omega} \lambda(\varphi(y) - f \log \varphi(y)) + \frac{1}{2} \sum_{s \in I} \| R_{\Omega_s} y - D\alpha_s \|_2^2 + \gamma \int_{\Omega} |\nabla y| \]

We state the proposed minimization problem in the following general form.

\[
\min_y F(y), \quad F(y) = f_1(y) + f_2(y)
\]

where \( f_1 \) with range \((-\infty, +\infty]\) is a proper, convex, differentiable function with a \(1/\beta\)-Lipschitz continuous gradient, and \( f_2 \) with range in \(\mathbb{R} \) is a proper, convex, lower-continuous function.
Forward-Backward algorithm

A forward-backward splitting process in optimization, consists of two separate steps:

\[
\begin{aligned}
    y_{n+\frac{1}{2}} &= \text{prox}_{\beta f_2} y_n & \text{forward step} \\
    y_{n+1} &= y_{n+\frac{1}{2}} - \beta \nabla f_1(y_{n+\frac{1}{2}}) & \text{backward step}
\end{aligned}
\]

If $\beta$ small enough, it converges.
Proposition

The $f_1$ given in (6) is Lipschitz differentiable with some Lipschitz constant $1/\beta$:

$$\|\nabla f_1(y_1) - \nabla f_1(y_2)\| \leq \frac{1}{\beta} \|y_1 - y_2\|.$$ 

Special case: $\beta = 0.01$ is enough.
Full algorithm for Poisson noise removal

Preprocess:

- Compute VST: $\phi(f)$.

Dictionary learning:

- Learning a dictionary $D$ from all the patches $R_{\Omega_s}\phi(f)$ using K-SVD.
- Represent $R_{\Omega_s}\phi(f)$ sparsely in the dictionary $D$ by OMP to get a denoised version: $D\alpha_s$.
- Compute $W = \sum_{s \in I} R^T_{\Omega_s} R_{\Omega_s}, M = \sum_{s \in I} R^T_{\Omega_s} D\alpha_s$.

Reconstruct the image:

- Solve the optimal minimization problem via a forward-backward splitting algorithm.
- The estimated image is $u = \varphi(y)$. 
Comparison

Figure: From left to right: Noisy image, TVMM, TVP, MSVST and Our algorithm.
Comparison

Figure: From left to right: Noisy image, TVMM, TVP, MSVST and Our algorithm.
Comparison

Figure: From left to right: Noisy image, TVMM, TVP, MSVST and Our algorithm.
Learned dictionary

(a) Barbara

(b) Cameraman

Figure: Barbara and Cameraman with peak intensity 30.
<table>
<thead>
<tr>
<th>Img/P_Int.</th>
<th>noisy</th>
<th>TVMM</th>
<th>TVP</th>
<th>MSVST</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bar./15</td>
<td>14.99</td>
<td>23.07</td>
<td>23.26</td>
<td>23.12</td>
<td>25.16</td>
</tr>
<tr>
<td>Bar./30</td>
<td>17.98</td>
<td>24.20</td>
<td>24.33</td>
<td>24.39</td>
<td>27.88</td>
</tr>
<tr>
<td>Bar./100</td>
<td>23.21</td>
<td>27.24</td>
<td>26.87</td>
<td>28.28</td>
<td>31.39</td>
</tr>
<tr>
<td>Bar./255</td>
<td>27.13</td>
<td>30.07</td>
<td>28.89</td>
<td>31.33</td>
<td>33.76</td>
</tr>
<tr>
<td>Bab./15</td>
<td>14.27</td>
<td>20.93</td>
<td>21.38</td>
<td>20.45</td>
<td>21.64</td>
</tr>
<tr>
<td>Bab./30</td>
<td>17.26</td>
<td>22.34</td>
<td>22.62</td>
<td>21.73</td>
<td>23.42</td>
</tr>
<tr>
<td>Bab./100</td>
<td>22.51</td>
<td>24.98</td>
<td>25.30</td>
<td>24.95</td>
<td>26.62</td>
</tr>
<tr>
<td>Bab./255</td>
<td>26.14</td>
<td>26.98</td>
<td>27.75</td>
<td>27.65</td>
<td>29.03</td>
</tr>
<tr>
<td>Boa./15</td>
<td>14.72</td>
<td>25.36</td>
<td>25.55</td>
<td>25.63</td>
<td>25.76</td>
</tr>
<tr>
<td>Boa./30</td>
<td>17.73</td>
<td>26.82</td>
<td>27.23</td>
<td>27.06</td>
<td>27.91</td>
</tr>
<tr>
<td>Boa./100</td>
<td>22.96</td>
<td>29.50</td>
<td>29.77</td>
<td>29.77</td>
<td>30.99</td>
</tr>
<tr>
<td>Boa./255</td>
<td>26.98</td>
<td>31.76</td>
<td>31.70</td>
<td>31.90</td>
<td>33.15</td>
</tr>
<tr>
<td>Img/P_Int.</td>
<td>noisy</td>
<td>TVMM</td>
<td>TVP</td>
<td>MSVST</td>
<td>Ours</td>
</tr>
<tr>
<td>-----------</td>
<td>-------</td>
<td>------</td>
<td>-----</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>Cam./15</td>
<td>15.05</td>
<td>24.50</td>
<td>24.74</td>
<td>25.09</td>
<td>25.72</td>
</tr>
<tr>
<td>Cam./30</td>
<td>18.01</td>
<td>26.21</td>
<td>26.64</td>
<td>26.53</td>
<td>27.98</td>
</tr>
<tr>
<td>Cam./100</td>
<td>23.26</td>
<td>29.17</td>
<td>29.31</td>
<td>29.41</td>
<td>30.98</td>
</tr>
<tr>
<td>Cam./255</td>
<td>27.31</td>
<td>31.86</td>
<td>31.39</td>
<td>31.73</td>
<td>33.57</td>
</tr>
<tr>
<td>Hou./15</td>
<td>14.17</td>
<td>26.57</td>
<td>26.10</td>
<td>27.14</td>
<td>27.32</td>
</tr>
<tr>
<td>Hou./30</td>
<td>17.16</td>
<td>28.22</td>
<td>28.39</td>
<td>28.78</td>
<td>30.01</td>
</tr>
<tr>
<td>Hou./100</td>
<td>22.42</td>
<td>30.94</td>
<td>31.61</td>
<td>31.34</td>
<td>33.25</td>
</tr>
<tr>
<td>Hou./255</td>
<td>26.15</td>
<td>32.92</td>
<td>33.14</td>
<td>33.35</td>
<td>35.02</td>
</tr>
</tbody>
</table>
Thank you!