Regularization by Denoising: Clarifications and New Interpretations

Phil Schniter and Ted Reehorst

The Ohio State University

With support from: NSF CCF-1527162 and NSF CCF-1716388

SIAM Conference on Imaging Science (Bologna, Italy) — June 7, 2018
Outline

- Introduction to RED
- Clarifications on RED
- New Interpretation of RED
- Fast RED Algorithms
Inverse Problems in Imaging

Inverse problems in imaging:

Recover $x^0$ from measurements $y = \text{corrupted}(Ax^0)$

where $A$ is a known linear operator.

In this talk, we’ll focus on additive white Gaussian noise (AWGN):

Recover $x^0$ from measurements $y = Ax^0 + e$ with $e \sim \mathcal{N}(0, \sigma^2 I)$.

Other corruptions include loss of phase, quantization, Poisson arrivals...
The variational approach to recovering $x$ solves an optimization problem:

$$\hat{x} = \arg\min_x \{ \ell(x; y) + \lambda \rho(x) \}$$

with

- $\ell(x; y)$: loss function
- $\rho(x)$: regularization
- $\lambda > 0$: tuning parameter

Can be interpreted as Bayesian MAP estimation:

$$\hat{x}_{\text{map}} = \arg\min_x \{- \ln p(y|x) - \ln p(x)\}$$

with

- $p(y|x)$: likelihood
- $p(x)$: prior

AWGN likelihood implies quadratic loss $\ell(x; y) = \frac{1}{2\sigma^2} \|Ax - y\|^2$.

But how should we choose the regularization $\rho(\cdot)$?
Recently, Romano, Elad and Milanfar\textsuperscript{1} proposed the RED regularization
\[ \rho_{\text{red}}(x) \triangleq \frac{1}{2} x^\top (x - f(x)) , \]
where \( f : \mathbb{R}^N \to \mathbb{R}^N \) is an image denoising function (e.g., BM3D).

RED leads to a family of “plug-and-play” (PnP) algorithms, similar to those proposed by Bouman et al.\textsuperscript{2} and Metzler et al.\textsuperscript{3}, but with some advantages.

\textsuperscript{1}Romano, Elad, Milanfar’17, \textsuperscript{2}Venkatakrishnan, Bouman, Wolhberg’13, \textsuperscript{3}Metzler, Maleki, Baraniuk’15
RED versus PnP

Experiments in the RED paper\textsuperscript{1} show advantages of RED algs over PnP:

Above represents super-resolution recovery averaged over 10 test images.
Claims about RED

The RED paper\(^1\) claims . . .

1. If \( f(\cdot) \) is **locally homogeneous** (LH), i.e.,
   \[
   f((1 + \epsilon)x) = (1 + \epsilon)f(x) \quad \text{for small } \epsilon,
   \]

   and **differentiable**, then gradient of \( \rho_{\text{red}}(x) \equiv \frac{1}{2} x^\top (x - f(x)) \) obeys
   \[
   \nabla \rho_{\text{red}}(x) = x - f(x).
   \]

2. If the Jacobian \( Jf(x) \) is **strongly passive**, i.e.,
   \[
   \| Jf(x) \|_2 \leq 1,
   \]

   then the RED regularization \( \rho_{\text{red}}(x) \) is **convex**.
Implications of RED Claims

- The convexity claim on $\rho_{\text{red}}(\cdot)$ implies that minimization of
  \[ C_{\text{red}}(x) \triangleq \frac{1}{2\sigma^2} \| Ax - y \|^2 + \lambda \rho_{\text{red}}(x) \]
can be easily tackled by many algs (e.g., SD, ADMM, etc.).

- The gradient claim $\nabla \rho_{\text{red}}(x) = x - f(x)$ implies the minimizers obey
  \[
  \text{RED fixed-point condition: } \frac{1}{\sigma^2} A^T (A\hat{x} - y) + \lambda (\hat{x} - f(\hat{x})) = 0
  \]
The RED algorithms find exactly these $\hat{x}$.
Introduction to RED

Mysterious Behavior

Surprisingly, the RED algorithms do not always behave as expected!

We expect SD to decrease the (convex) RED cost, but it is increasing it!

\[
\text{RED-SD: } \mathbf{x}_{k+1} = \mathbf{x}_k - \mu \nabla C_{\text{red}}(\mathbf{x}_k)
\]
Clarifications on RED Gradient

It can be shown that...

- **differentiability** in $f(\cdot)$ implies

  $$\nabla \rho_{\text{red}}(x) \overset{D}{=} x - \frac{1}{2} f(x) - \frac{1}{2} [J f(x)]^\top x.$$  

- adding **local-homogeneity** (LH) gives

  $$\nabla \rho_{\text{red}}(x) \overset{D,\text{LH}}{=} x - \frac{1}{2} [J f(x)] x - \frac{1}{2} [J f(x)]^\top x.$$  

- adding **Jacobian symmetry** (JS) finally leads to

  $$\nabla \rho_{\text{red}}(x) \overset{D,\text{LH,JS}}{=} x - f(x) \ldots \text{which yields the RED algorithms.}$$

So **both LH and JS** are needed to link RED cost to RED alg.
Which Denoisers Yield Jacobian Symmetry?

Clear that these yield JS:
- Linear denoisers $f(x) = Wx$ with $W = W^\top$.
- Transform-domain-thresholding (TDT) denoisers $f(x) = W^\top g(Wx)$.
- MAP or MMSE denoisers under any assumed prior $x \sim \hat{p}_x$.

Not clear that these yield JS:
- Pseudo-linear denoisers $f(x) = W(x)x$ with non-linear $W(\cdot)$.
- Approximately MAP or MMSE denoisers.

Most state-of-the-art denoisers fall into the 2nd category.
### Jacobian Symmetry Experiments

**Avg JS error on suite of $16 \times 16$ images:**

<table>
<thead>
<tr>
<th></th>
<th>TDT</th>
<th>MF</th>
<th>NLM</th>
<th>BM3D</th>
<th>TNRD</th>
<th>DnCNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{</td>
<td></td>
<td>\hat{J}f(x) - [\hat{J}f(x)]^\top</td>
<td></td>
<td>_F^2}{</td>
<td></td>
<td>\hat{J}f(x)</td>
</tr>
</tbody>
</table>

**Avg gradient error on suite of $16 \times 16$ images:**

<table>
<thead>
<tr>
<th></th>
<th>TDT</th>
<th>MF</th>
<th>NLM</th>
<th>BM3D</th>
<th>TNRD</th>
<th>DnCNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{</td>
<td></td>
<td>\nabla \rho_{\text{red}}(x) - \hat{\nabla} \rho_{\text{red}}(x)</td>
<td></td>
<td>_2^2}{</td>
<td></td>
<td>\nabla \rho_{\text{red}}(x)</td>
</tr>
<tr>
<td>$\nabla \rho_{\text{red}}(x)$ with D</td>
<td>0.565</td>
<td>0.966</td>
<td>0.913</td>
<td>1.00</td>
<td>0.957</td>
<td>0.852</td>
</tr>
<tr>
<td>$\nabla \rho_{\text{red}}(x)$ with D,LH,JS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Key points:**

1. **Large JS error** for all but TDT.
2. **Large gradient error** under JS & LH assumptions for all denoisers!
3. Even TDT has large gradient error! Is LH the problem?
Local Homogeneity Experiments

Avg LH error on suite of $16 \times 16$ images:

<table>
<thead>
<tr>
<th></th>
<th>TDT</th>
<th>MF</th>
<th>NLM</th>
<th>BM3D</th>
<th>TNRD</th>
<th>DnCNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>$|f((1+\epsilon)x-(1+\epsilon)f(x))|^2 |\epsilon f(x)|^2$</td>
<td>7.99e-10</td>
<td>0</td>
<td>5.60e-9</td>
<td>1.52e-13</td>
<td>5.09e-10</td>
<td>2.06e-9</td>
</tr>
<tr>
<td>$|\hat{J}_f(x)\cdot x-f(x)|^2 |f(x)|^2$</td>
<td>4.10e-4</td>
<td>2.14e-15</td>
<td>5.63e-3</td>
<td>0.214</td>
<td>2.60e-4</td>
<td>8.02e-3</td>
</tr>
</tbody>
</table>

Avg gradient error on suite of $16 \times 16$ images:

<table>
<thead>
<tr>
<th></th>
<th>TDT</th>
<th>MF</th>
<th>NLM</th>
<th>BM3D</th>
<th>TNRD</th>
<th>DnCNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>$|\nabla\rho_{\text{red}}(x) - \hat{\nabla}\rho_{\text{red}}(x)|^2 |\hat{\nabla}\rho_{\text{red}}(x)|^2$</td>
<td>3.39e-19</td>
<td>2.65e-15</td>
<td>6.17e-21</td>
<td>2.14e-13</td>
<td>5.42e-17</td>
<td>1.02e-12</td>
</tr>
<tr>
<td>$\nabla\rho_{\text{red}}(x)$ with D</td>
<td>0.565</td>
<td>6.09e-15</td>
<td>0.0699</td>
<td>0.344</td>
<td>0.139</td>
<td>1.20</td>
</tr>
<tr>
<td>$\nabla\rho_{\text{red}}(x)$ with D, LH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Key points:

- It is important how LH is quantified.
- The RED gradient is very sensitive to small imperfections in LH.
Implications of our Findings

We found:

- The RED algorithms solve a fixed-point equation corresponding to \( \nabla \rho(x) = x - f(x) \).
- \( x - f(x) \) is very different from \( \nabla \rho_{\text{red}}(x) \) under practical \( f(\cdot) \), such as TDT, MF, NLM, BM3D, TNRD, and DnCNN.

Implication:

- \( \rho_{\text{red}}(\cdot) \) does not explain the RED algorithms under practical \( f(\cdot) \).

A bigger problem:

- For non-JS \( f(\cdot) \), can show that there exists no explicit regularizer \( \rho(\cdot) \) for which \( \nabla \rho(x) = x - f(x) \), i.e., explaining the RED algorithms!
How to Explain the RED Algorithms?

The RED algorithms assume $\nabla \rho(x) = x - f(x)$ and work very well.

Can we justify this $\nabla \rho(x)$?
Even when $f(\cdot)$ is not LH and/or JS?

Yes! Using score matching. We explain this in 3 steps:

1. regularization by log-likelihood (RLL),
2. RLL as kernel density estimation (KDE),
3. score matching.
Consider noisy pseudo-measurements

\[ r = x^0 + \mathcal{N}(0, \nu I) \]

Suppose we adopt the prior pdf \( \hat{p}_x \). Then the likelihood of \( r \) is

\[ \hat{p}_r(r; \nu) = \int_{\mathbb{R}^N} \mathcal{N}(r; x, \nu I) \hat{p}_x(x) \, dx. \]

“Gaussian blurred prior”

Define the RLL regularization as

\[ \rho_{LL}(r; \nu) \triangleq -\nu \ln \hat{p}_r(r; \nu) \]

Then it can be shown using Tweedie’s formula\(^4\) that

\[ \nabla \rho_{LL}(r; \nu) = r - \hat{f}_{\text{mmse}, \nu}(r), \]

which is consistent with the RED algorithms!

---

\(^4\)Robbins’56
RLL as Kernel Density Estimation

- Given training data \( \{x_t\}_{t=1}^T \), consider the empirical prior

\[
\hat{p}_x(x) = \frac{1}{T} \sum_{t=1}^{T} \delta(x - x_t).
\]

- A better match to the true \( p_x \) is obtained via KDE or Parzen windowing:

\[
\tilde{p}_x(x; \nu) = \frac{1}{T} \sum_{t=1}^{T} \mathcal{N}(x; x_t, \nu I).
\]

  “blurred empirical prior”

- Using this \( \tilde{p}_x \) for MAP/variational optimization yields

\[
\hat{x} = \arg \min_x \frac{1}{2\sigma^2} \|Ax - y\|^2 - \ln \tilde{p}_x(x; \nu)
= \arg \min_x \frac{1}{2\sigma^2} \|Ax - y\|^2 + \lambda \rho_{\text{LL}}(x; \nu) \text{ for } \lambda = \frac{1}{\nu}.
\]

So RLL arises naturally in non-parametric estimation via KDE!
The above RLL/KDE framework encompasses only JS denoisers $f(\cdot)$. We now generalize.

First note that, for large $T$, gradient is very expensive:

$$
\nabla \ln \tilde{p}_x(x;\nu) = \frac{\hat{f}_{\text{mmse},\nu}(x) - x}{\nu} \quad \text{with} \quad \hat{f}_{\text{mmse},\nu}(x) = \frac{\sum_{t=1}^{T} (x_t - x) N(x; x_t, \nu I)}{\sum_{t=1}^{T} N(x; x_t, \nu I)}.
$$

Practical idea:\footnote{Hyvärinen’05} use best match to “score” $\nabla \ln \tilde{p}_x(x)$ among computationally friendly functions $\psi(x;\theta)$:

$$
\hat{\theta} = \arg \min_{\theta} \mathbb{E}_{\tilde{p}_x} \{ \| \psi(x;\hat{\theta}) - \nabla \ln \tilde{p}_x(x;\nu) \|^2 \}.
$$

Vincent\footnote{Vincent’11} connected to denoising: if $\psi(x;\theta) = [f(x;\theta) - x]/\nu$, then

$$
\hat{\theta} = \arg \min_{\theta} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E} \{ \| f_{\theta}(x_t + N(0, \nu I)) - x_t \|^2 \},
$$

where $f_{\hat{\theta}}(\cdot)$ is MMSE optimal $f_{\theta} \in \mathcal{F}$, where $\mathcal{F} \triangleq \{ f_{\theta} : \theta \in \Theta \}$.
Score-Matching by Denoising (SMD)

Key points:

1. SMD interpretation yields $\nabla \rho(x) = x - f(x)$, thus explaining RED algs.

2. SMD interpretation holds for any $\hat{p}_x$, any denoiser class $F$ (i.e., $f_\theta$ may be non-JS and/or non-LH), and any $\theta$ (maybe not MMSE).

3. SMD arises naturally via non-parametric estimation and KDE. Matches construction of learned denoisers liked TNRD and DnCNN.

Related work:
Alain and Bengio\textsuperscript{7} recently showed that learned auto-encoders can be explained by score-matching and \textit{not} by minimization of an energy function.

\textsuperscript{7}Alain/Bengio’14
Fast RED Algorithms

Until now we focused on how to explain the RED algorithms, which solve

\[
\text{RED fixed-point condition: } \quad \frac{1}{\sigma^2} A^\top (A\hat{x} - y) + \lambda(\hat{x} - f(\hat{x})) = 0
\]

We now focus on interpretation/design of fast RED algorithms.

In the RED paper, three algorithms were described:

1. Steepest-Descent
2. ADMM with \( I \) inner iters (to solve \( \arg\min_x \{ \lambda \rho(x) + \frac{\beta}{2} \| x - r_k \|_2^2 \} \))
3. A “fixed-point” method (we show equivalence to proximal gradient alg\(^8\))

We propose a couple more...

\(^8\) Combettes/Pesquet’11
New algorithms:

- **DPG**: “Dynamic” proximal gradient, which schedules the stepsize.

- **APG**: Accelerated proximal gradient, similar to FISTA.\(^9\)

In this experiment, APG is about \(3 \times\) faster than the Fixed-Point method.

\(^9\)Beck/Teboulle’09
The RED algorithms work very well in practice.

But they do not minimize $C_{\text{red}}(x) = \ell(x; y) + \lambda \rho_{\text{red}}(x)$ for many $f(\cdot)$.

- Why? Practical denoisers $f(\cdot)$ are not sufficiently LH and JS.
- Can construct examples of RED-SD *increasing* $C_{\text{red}}(x)$ over the iterations.

We explained RED algorithms as “score-matching by denoising”.

We proposed new RED algorithms with faster convergence.