SOS Boosting of Image Denoising Algorithms*

Michael Elad
The Computer Science Department
The Technion – Israel Institute of technology
Haifa 32000, Israel

* Joint work with Yaniv Romano

The research leading to these results has received funding from the European Research Council under European Union's Seventh Framework Program, ERC Grant agreement no. 320649, and by the Intel Collaborative Research Institute for Computational Intelligence.
Image Denoising

Probably the most popular and heavily studied problem in image processing ...

Why is it so popular? Here are few explanations:
(i) It does come up in many applications
(ii) It is the simplest inverse problem, platform for new ideas
(iii) Many other problems can be recast as an iterated denoising, and ...
(iv) It is misleadingly simple

Searching image and (denois* or (noise and removal)) in ISI Web-of-Science, leads to ~5000 journal papers
Leading Image Denoising Methods

Are built upon powerful patch-based (local) image models:

- Non-Local Means (NLM): self-similarity within natural images
- K-SVD: sparse representation modeling of image patches
- BM3D: combines a sparsity prior and non local self-similarity
- Kernel-regression: offers a local directional filter
- EPLL: exploits a GMM model of the image patches
- ...

Today we present a way to improve various such state-of-the-art image denoising methods, simply by applying the original algorithm as a “black-box” several times.
Background
In image denoising, there are two sources of possible problems:

- Residual noise in the output image, and
- Residual content in the method noise

\[ \hat{x} = f(y) \]
Existing Boosting Algorithms

- **Twicing** [Tukey (’77), Charest et al. (’06)]
  \[ \hat{x}^{k+1} = \hat{x}^k + f(y - \hat{x}^i) \]

- **TV denoising using Bregman distance** [Bregman (’67), Osher et al. (’05)]
  \[ \hat{x}^{k+1} = f(\hat{x}^k + \sum_{i=1}^{k}(y - \hat{x}^i)) \]

SAIF [Talebi et al. (’12)] chooses automatically the local improvement mechanism: Diffusion or Twicing

- **EPLL** [Zoran & Weiss (’09), Sulam & Elad (’14)]
SOS Boosting

Boosting of Image Denoising Algorithms
SIAM Journal on Imaging Sciences, 2015
Given any denoiser, how can we improve its performance?
Given any denoiser, how can we improve its performance?

I. **Strengthen** the signal

II. **Operate** the denoiser
Given any denoiser, how can we improve its performance?

I. **Strengthen** the signal
II. **Operate** the denoiser
III. **Subtract** the previous estimation from the outcome

**SOS formulation:** \( \hat{x}^{k+1} = f(y + \hat{x}^k) - \hat{x}^k \)
Strengthen - Operate - Subtract Boosting

- An improvement is expected since \( \text{SNR}\{y + \hat{x}\} > \text{SNR}\{y\} \)

  In the ideal case, where \( \hat{x} = x \), we get

  \[
  \text{SNR}\{y + x\} = 2 \cdot \text{SNR}\{y\}
  \]

- We suggest strengthening the underlying signal, rather than
  - Adding/filtering the method noise – which tends to converge to the noisy image, or
  - Operating on the denoising output again and again – which tends to lead to over-smoothing

- SOS treats both sources of errors created in image denoising ...
Observation:

We study the convergence of the SOS using only the linear part:

$$\hat{x} = f(y) = W y$$

What about sparsity-based denoising methods [Elad & Aharon ('06)]? We have shown that in this case that

- $W$ is symmetric and positive definite,
- and $\lambda_{min} \geq c > 0$, and $\lambda_{max} = 1$, 

$$\|W - I\|_2 \leq 1 - c < 1$$

Denoising Algorithm

Non-Linear Part (decisions/switches)

Spatially adaptive weighted averages

True for NLM, Kernel-regression, BM3D, K-SVD, and many other methods
Convergence Study

Theorem: The SOS converges if $\|I - W\|_2 < 1$, which holds true for kernel-based (Bilateral filter, NLM, Kernel Regression), and sparsity-based methods (K-SVD).

For these denoising algorithms, the SOS boosting converges to

$$\hat{x}^* = (I + (I - W))^{-1}Wy$$

What about the non-linear part and its influence? More on this can be found in our paper...
Generalization

- We introduce two parameters that modify
  - The steady-state outcome
  - The requirements for convergence (the eigenvalues range), and
  - The rate of convergence

- The parameter $\rho$, affects the steady-state outcome:

$$\hat{x}^{k+1} = f(y + \rho \hat{x}^k) - \rho \hat{x}^k$$

- The second parameter, $\tau$, controls the rate-of-convergence, without affecting the steady-state:

$$\hat{x}^{k+1} = \tau f(y + \rho \hat{x}^k) - (\tau \rho + \tau - 1) \hat{x}^k$$
Generalization

- By defining the error \( e_k = \hat{x}^k - \hat{x}^* \), the SOS yields:

\[
e_k = (\tau \rho W - (\tau \rho + \tau - 1)I) e_{k-1}
\]

- We derived a closed-form expression for the optimal \((\rho, \tau)\) setting
  - Given \(\rho\), what is the best \(\tau\), leading to the fastest convergence? This is depicted by the dashed curve.

Largest eigenvalue of the error’s transition matrix

---

Boosting of Image Denoising Algorithms
By Yaniv Romano and Michael Elad
Graph-Based Interpretation
Graph Laplacian

- We can refer to the denoising matrix, $W$, as a spatially adaptive smoothing operator.
- A Graph Laplacian operator for an image can be defined as $\mathcal{L} = I - W$.
  
  The intuition: The Laplacian computes the difference between a pixel and its neighborhood’s weighted average.
- The image content is expected to reside along the eigenvectors corresponding to the small eigenvalues of $W$, while the noise is spread uniformly over all the eigenspace.
- What can we do with $\mathcal{L}$? $\rightarrow$ Regularize inverse problems!
The regularization can be defined as [Elmoataz et al. ('08), Bougleux et al. ('09)]

\[
\hat{x} = \min_x \|x - y\|_2^2 + \rho x^T \mathcal{L} x
\]

Seeks for an estimation that is close to the noisy version

While promoting similar pixels to remain similar

Another option is to integrate the filter also in the data fidelity term [Kheradmand and Milanfar ('13)]

\[
\hat{x} = \min_x (x - y)^T W (x - y) + \rho x^T \mathcal{L} x
\]

Using the adaptive filter as a weight-matrix
Graph Laplacian Regularization

Another natural option is to minimize the following cost function:

$$\hat{x}^* = \min_{x} \|x - Wy\|^2_2 + \rho x^T \mathcal{L} x$$

This seeks for an estimation that is close to the denoised version.

Its closed-form solution is the steady-state outcome of the SOS:

$$\hat{x}^* = (I + \rho(I - W))^{-1}Wy = (I + \rho \mathcal{L})^{-1}Wy$$

The SOS boosting acts as a graph Laplacian regularizer.

More on this topic can be found in our paper ...
Experiments
Results

- We successfully boost several state-of-the-art denoising algorithms:
  - K-SVD, NLM, BM3D, and EPLL
  - Without any modifications, simply by applying the original software as a “black-box”

- We manually tuned two parameters:
  - $\rho$ – signal emphasis factor
    \[
    \hat{x}^{k+1} = f_{\sigma}(y + \rho\hat{x}^k) - \rho\hat{x}^k
    \]
  - $\sigma$ – noise level, which is an input to the denoiser
    - Since the noise level of $y + \rho x^k$ is higher than the one of $y$
## Quantitative Comparison

- **Average boosting in PSNR* over 5 images (higher is better):**

<table>
<thead>
<tr>
<th>Noise std</th>
<th>Improved Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>K-SVD</td>
</tr>
<tr>
<td>$\sigma$</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.13</td>
</tr>
<tr>
<td>20</td>
<td>0.22</td>
</tr>
<tr>
<td>25</td>
<td>0.26</td>
</tr>
<tr>
<td>50</td>
<td>0.77</td>
</tr>
<tr>
<td>75</td>
<td>1.26</td>
</tr>
<tr>
<td>100</td>
<td>0.81</td>
</tr>
</tbody>
</table>

*PSNR = 20\log_{10}(255/\sqrt{MSE})*
Visual Comparison: K-SVD

Original K-SVD results, $\sigma = 25$

29.06dB
Visual Comparison: K-SVD

- **SOS** K-SVD results, $\sigma = 25$

29.41 dB
Visual Comparison: EPLL

- Original EPLL results, $\sigma = 25$

Forman

32.44dB

House

32.07dB
Visual Comparison: EPLL

- **SOS EPLL results, \( \sigma = 25 \)**

  - **Forman**
    - 32.78 dB
  - **House**
    - 32.38 dB

Boosting of Image Denoising Algorithms
By Yaniv Romano and Michael Elad
Visual Comparison: All

Noisy image                    KSVD (31.20)                NLM (30.02)               BM3D (31.88)          EPLL (30.88)

SOS KSVD (31.91)       SOS NLM (30.56)       SOS BM3D (31.94)      SOS EPLL (31.15)
Visual Comparison: All

Noisy image  KSVD (33.72)  NLM (31.64)  BM3D (34.66)  EPLL (33.62)

SOS KSVD (34.4)  SOS NLM (32.3)  SOS BM3D (34.7)  SOS EPLL (34.1)
Time to Conclude
Conclusions

The SOS boosting algorithm is:

✓ Easy to use - we simply treat the denoiser \( f(\cdot) \) as a “black-box”
✓ Applicable to a wide range of denoising algorithms \( f(\cdot) \)
✓ Guaranteed to converge for many leading denoising algorithms
✓ Has a straightforward stopping criterion
✓ Acts as an interesting graph-Laplacian regularizer
✓ Reduces the local-global gap in patch-based methods
✓ Guaranteed to improve state-of-the-art methods
✓ Has an automatic parameter-settings based on MC-SURE
We are Done...

Thank you!

Questions?
K-SVD Denoising
Matrix Formulation
We assume the existence of a dictionary $D \in \mathbb{R}^{d \times n}$ whose columns are the atom signals.

Signals are modeled as sparse linear combinations of the dictionary atoms:

$$x = D\alpha$$

where $\alpha$ is sparse, meaning that it is assumed to contain mostly zeros.

The computation of $\alpha$ from $x$ (or its or its noisy version) is called sparse-coding.

The OMP is a popular sparse-coding technique, especially for low dimensional signals.
K-SVD Image Denoising

[Elad & Aharon (‘06)]

Noisy Image

Initial Dictionary

Using KSVD

Update the Dictionary

Using OMP

Denoise each patch

\[
\hat{p}_i = D_{S_i} \alpha_i = D_{S_i} \left( D_{S_i}^T D_{S_i} \right)^{-1} D_{S_i}^T R_i y
\]

\( \alpha_i = \min_z \| D_{S_i} z - R_i y \|_2^2 \)

\( R_i \) extracts the \( i^{th} \) patch from \( y \)

Denoised Patch

A linear combination of few atoms

Boosting of Image Denoising Algorithms
By Yaniv Romano and Michael Elad
K-SVD Image Denoising  

Noisy Image  

Initial Dictionary  

Using OMP  

Denoise each patch  

Using KSVD  

Update the Dictionary  

Reconstructed Image  

\[
\hat{x} = \min_{x} \mu \| x - y \|_2^2 + \sum_i \| \hat{p}_i - R_i x \|_2^2 \\
= \left( \mu I + \sum_i R_i^T R_i \right)^{-1} \left( \mu I + \sum_i R_i^T D_{S_i} \left( D_{S_i}^T D_{S_i} \right)^{-1} D_{S_i}^T R_i \right)y \\
= W y
\]
Reducing the “Local-Global” Gap
It turns out that the SOS boosting reduces the local/global gap, which is a shortcoming of many patch-based methods:

- **Local processing of patches VS. the global need** in a whole denoised result

We define the local disagreements by:

- Naturally exist since each noisy patch is denoised independently
- Are based on the **intermediate** results
“Sharing the Disagreement”

- Inspired by the “Consensus and Sharing” problem from game-theory:
  - There are several agents, each one of them aims to minimize its individual cost (i.e., representing the noisy patch sparsely)
  - These agents affect a shared objective term, describing the overall goal (i.e., obtaining the globally denoised image)

- Imitating this concept, we suggest sharing the disagreements

Noisy patches → + → Patch-based denoising → Patch Avg. → Est. Image

Noisy image → − → Disagreement per-patch → + → Patch Avg. → − → Est. Image

Boosting of Image Denoising Algorithms
By Yaniv Romano and Michael Elad
Connection to SOS Boosting

- Interestingly, for a fixed filter matrix $W$, “sharing the disagreement” and the SOS boosting are equivalent

$$\hat{x}^{k+1} = W(y + \hat{x}^k) - \hat{x}^k$$

- The connection to the SOS is far from trivial because
  - The SOS is blind to the intermediate results (the independent denoised patches, before patch-averaging)
  - The intermediate results are crucial for “sharing the disagreement” approach

The SOS boosting reduces the local-global gap
Guaranteed improvement?
The General Case [Milanfar (’12)]

- Given any denoiser $\mathbf{W}$, the MSE of $\hat{x} = \mathbf{W}y$ can be expressed by

$$\text{MSE}(\hat{x}) = \frac{1}{N} \|x - \hat{x}\|_2^2 = \|\text{bias}(\hat{x})\|_2^2 + \text{var}(\hat{x})$$

  - $\text{bias}(\hat{x}) = E[\hat{x}] - x = (\mathbf{W} - \mathbf{I})x$
  - $\text{var}(\hat{x}) = E[(\hat{x} - E[\hat{x}])^2] = \sigma_y^2 \cdot \text{Tr}\{\mathbf{W}\mathbf{W}^T\}$

- We define the eigen-decomposition by

$$\mathbf{W} = \mathbf{V}\Lambda\mathbf{V}^T$$

  - $\mathbf{V}$: orthogonal matrix, representing the “denoiser space”
  - $\Lambda$: diagonal matrix, containing the eigenvalues $\lambda_i$
The General Case [Milanfar (’12)]

- Setting \( x = \mathbf{V} \mathbf{b} \), we get
  \[
  \text{MSE}(\hat{x}) = \sum_{i=1}^{N} (\lambda_i - 1)^2 b_i^2 + \sigma_v^2 \sum_{i=1}^{N} \lambda_i^2
  \]
  
  - Tradeoff: \( \lambda \to 1 \) reduces the bias but increases the variance

- The optimal filter (\( \nabla_{\lambda_i} \text{MSE}(\hat{x}) = 0 \)) is obtained for
  \[
  \lambda_i^{\text{opt}} = \frac{b_i^2}{b_i^2 + \sigma_v^2}
  \]
  which is the Wiener filter, but it requires knowledge of \( b_i^2 \)
MSE of the SOS boosting

- In the case of the SOS boosting, the filter can be represented as

\[
W^{sos} = \left( I + \rho (I - W) \right)^{-1} W = V\tilde{\Lambda}V^T
\]

where \( \tilde{\lambda}_i = \frac{1}{1 + \rho(1 - \lambda_i)} \)

- As a result, for \( \rho > 0 \), we get

\[
\text{MSE}(x^{sos}) = \sum_{i=1}^{N} \left( \frac{1 + \rho}{1 + \rho(1 - \lambda_i)} \right)^2 \text{bias}_i^2(\hat{x}) + \sum_{i=1}^{N} \left( \frac{1}{1 + \rho(1 - \lambda_i)} \right)^2 \text{var}_i(\hat{x})
\]

- A large \( \rho \) reduces the variance of \( \hat{x} \) but increases the bias of \( \hat{x} \)

Larger than 1  Smaller than 1
Gaining Improvement

Let us define an improvement function

$$\Phi(\rho) = \text{MSE}(x_{\rho}^{\text{SOS}}) - \text{MSE}(\hat{x})$$

- An improvement is obtained if $$\Phi(\rho) < 0$$

**Theorem:** For any denoiser $$W$$ with $$\lambda_i \neq \frac{b_i^2}{b_i^2 + \sigma_v^2}$$ (suboptimal eigenvalues), $$\exists \rho^*$$ such that $$\Phi(\rho^*) < 0$$

The SOS boosting is always able to improve a suboptimal denoiser
Minimizing the MSE

- We don’t have the true MSE
- But we can estimate it using SURE [Stein ('81)]

\[
\text{MSE} \left( f_\rho(y) \right) \approx \| f_\rho(y) \|^2_2 - 2f_\rho(y)^T y + 2\sigma_v^2 \cdot \text{Tr}\{\nabla_y f_\rho(y)\}
\]

- Requires the analytical form of the divergence of \( f_\rho(y) \)

- Solution: Monte-Carlo SURE [Ramani et al. ('08)]
  - Treats the denoiser as a black-box
  - Estimates the first order difference estimator of the divergence