Effective Snapshot Compressive-spectral Imaging via Deep Denoising and Total Variation Priors

Yao Wang
Center for Intelligent Decision-making and Machine Learning
Xi’an Jiaotong University

April 2022

Joint work with Haiquan Qiu and Prof. Deyu Meng
High-dimensional Image Data Everywhere

Hyperspectral

Surveillance

MRI

How to efficiently extract compact information from such complex data?
High-dimensional Sparse Modeling

Applying the high-dimensional sparse model

\[
\min_{\mathcal{X}} \ S(\mathcal{X}) \\
\text{s.t. } \ y = \phi(\mathcal{X}) + \epsilon
\]

Main issues
- Sampling mechanism
- Structure measure
- Recovery algorithm
- Performance assessment
- Applications

Cao, et al., TIP’16; Wang, et al., NSR’ 18; Wang, et al., TGRS’20; Peng, et al., TIP’20; Hou, et al., TPAMI’21
The term "remote sensing" generally refers to the use of satellite- or aircraft-based sensor technologies to detect and classify objects on Earth, including on the surface and in the atmosphere and oceans, based on propagated signals.
Compressive-spectral Imaging

Spectral image: a *three-dimensional* \((x,y,\lambda)\) data cube, where \(x\) and \(y\) represent two spatial dimensions of the scene, and \(\lambda\) represent the spectral of dimension.
The encoding process of SCI

Yuan, Brady and Katsaggelos, IEEE SPM’21
Snapshot Compressive Imaging

The decoding process of SCI

Yuan, Brady and Katsaggelos, IEEE SPM’21
The sensing ratio of SCI is $1/N_t$ (in single-pixel imaging, it is $M/N$)
CASSI System

The sensing process of CASSI (coded aperture compressive spectral imager)

Brady, et al., OE’09; Lin, et al., ACM TOG’14; Yuan, et al., IEEE JSTSP’15
Plug-and-Play Algorithms

- The mathematical model:

\[
\hat{x} = \arg \min_x \frac{1}{2} \| y - Hx \|_2^2 + \lambda g(x)
\]

- PnP-ADMM: (Ryu, et al., ICML’18; Chan, et al., IEEE TCI’17)

\[
x^{(k+1)} = (H^T H + \gamma I)^{-1} [H^T y + \gamma (v^{(k)} + u^{(k)})]
\]

\[
v^{(k+1)} = D_\sigma (x^{(k+1)} - u^{(k)})
\]

\[
u^{(k+1)} = u^{(k)} + (v^{(k+1)} - x^{(k+1)})
\]

- PnP-GAP: (Yuan, et al., CVPR’20, TPAMI’21)

\[
x^{(k+1)} = v^{(k)} + H^T (H H^T)^{-1} (y - H v^{(k)})
\]

\[
v^{(k+1)} = D_\sigma (x^{(k+1)})
\]

Key idea: replaces the proximal operator with a denoiser
TV+Pretrained FFDNet

- The variation along the spectral direction is *very small*
- It is typically *piecewise smooth* along the spatial domain
TV+Pretrained FFDNet

- Deep denoiser is efficient while some model-based methods such as BM3D and WNNM are time-consuming.
- Spatially variant noise can be flexibly handled.
- Learn to characterize complex image structures from other data.

Deep Denoising Prior
TV+Pretrained FFDNet

- FFDNet & TV
- Ours
- Best
The Proposed Procedure

- Treat the denoising step as MAP estimation:
  \[ v^{(k+1)} = \arg \min_v \frac{\lambda}{2} g(v) + \frac{1}{2} \| x^{(k+1)} - v \|_2^2 = \arg \max_v p(v | x^{(k+1)}, \sigma) = D_\sigma(x^{(k+1)}) \]

- Eliminate the hyperparameters as
  \[ p(v | x^{(k+1)}) = \int p(v | x^{(k+1)}, \sigma) p(\sigma | x^{(k+1)}) d\sigma \]
  \[ q(v | x^{(k+1)}) = \int q(v | x^{(k+1)}, t) q(t | x^{(k+1)}) dt \]

- Minimizing the distance:
  \[ \min_{p(\sigma | x^{(k+1)}), q(t | x^{(k+1)})} \text{dist} \left( p(v | x^{(k+1)}), q(v | x^{(k+1)}) \right) \]

Use Gaussian distribution to model the posterior and discrete technique
The Proposed Procedure

Algorithm 1 The proposed PnP-GAP

Require: $H, y$.
1: Initial $v^{(0)}$, $A$, $B$
2: while Not Converge do
3: Update $x$ by Eq. (9).
4: Obtain denoising image set $\{v_\sigma : \sigma \in A\}$ by $v^{(k+1)}_\sigma = \text{FFD}_\sigma(x^{(k+1)})$.
5: Obtain denoising image set $\{v_t : t \in B\}$ by $v^{(k+1)}_t = \text{TV}_t(x^{(k+1)})$.
6: Solve optimization problem (20).
7: Update $v$ by Eq. (21)
8: end while

$$x^{(k+1)} = v^{(k)} + HT(HT)^{-1}(y - Hv^{(k)}), \quad (9)$$

$$\begin{align*}
\min_{W} \quad & W^T PW \\
\text{subject to} \quad & \sum_{i=1}^{\vert A \vert+\vert B \vert} W_i = 1 \\
& \sum_{i=1}^{\vert A \vert} W_i = 1, W \geq 0,
\end{align*} \quad (20)$$

$$v^{(k+1)} = \frac{1}{2} \left( \sum_{\sigma \in A} \hat{w}_\sigma^{ff}v_\sigma + \sum_{t \in B} \hat{w}_t^{tv}v_t \right). \quad (21)$$

More details can be found in our CVPR21’s paper
Fixed-point Convergence

**Assumption 1.** We assume that all denoisers $D_\sigma : \mathbb{R}^d \mapsto \mathbb{R}^d$ used in our method satisfy

$$
\| (D_\sigma - I)(x) - (D_\sigma - I)(y) \|_2 \leq \epsilon \| x - y \|_2
$$

for all $x, y \in \mathbb{R}^d$ for some $\epsilon > 0$.

**Assumption 2.** Assume that $\{R_j\}_{j=1}^n > 0$ which means for each spatial location $j$, the $B$-frame modulation masks at this location have at least one non-zero entries. We further assume $R_{\text{max}} > R_{\text{min}}$.

**Theorem 1.** Assume $H$ satisfies Assumption 2. Then the following operator

$$
G = D_\sigma \circ P
$$

is a contraction if $D_\sigma$ satisfies Assumption 1 and

$$
0 < \epsilon < \sqrt{\frac{R_{\text{max}}}{R_{\text{max}} - R_{\text{min}}}} - 1.
$$

**Theorem 2.** Assume $H$ satisfies Assumption 2. Let $P$ be a Euclidean projection on linear manifold $y = Hx$. Then

$$
T = \frac{1}{2} I + \frac{1}{2} (2P - I)(2D_\sigma - I)
$$

is a contraction if $D_\sigma$ satisfies Assumption 1 and

$$
0 < \epsilon < 1 - \sqrt{1 - \frac{R_{\text{min}}}{R_{\text{max}}}}.
$$

We can also prove the convergence of PnP-ADMM.
Experiments

- **Bird** data consists of 24 spectral bands, and the size of each spectral band is $1021 \times 703$.
- **Toy** data consists of 31 bands, and the size of each band is $512 \times 512$.
- **CAVE** data includes 32 spectral images, and each image contains 31 spectral bands. The image size of each band is $512 \times 512$.
- Our code is available at [https://github.com/ucker/SCI-TVFFDNet](https://github.com/ucker/SCI-TVFFDNet).
Comparison with Non-DL Methods

Table 1. The results of PSNR in dB (left entry in each cell) and SSIM (right entry in each cell) by different algorithms on Bird and Toy.

<table>
<thead>
<tr>
<th>Data</th>
<th>2DTV</th>
<th>3DTV</th>
<th>FFDNet</th>
<th>DeSCI</th>
<th>FFDNet-TV</th>
<th>Ours (2DTV)</th>
<th>Ours (3DTV)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toy</td>
<td>25.26, 0.8630</td>
<td>28.46, 0.9102</td>
<td>24.28, 0.8298</td>
<td>26.62, 0.9116</td>
<td>25.49, 0.8748</td>
<td><strong>29.35, 0.9249</strong></td>
<td><strong>28.86, 0.9225</strong></td>
</tr>
<tr>
<td>Bird</td>
<td>37.58, 0.9361</td>
<td>25.84, 0.7919</td>
<td>36.60, 0.9171</td>
<td>38.25, 0.9520</td>
<td>38.21, 0.9383</td>
<td><strong>39.73, 0.9559</strong></td>
<td><strong>31.30, 0.9069</strong></td>
</tr>
</tbody>
</table>

Table 2. The average results of PSNR in dB (left entry in each cell) and SSIM (right entry in each cell) by different algorithms on CAVE.

<table>
<thead>
<tr>
<th></th>
<th>2DTV</th>
<th>3DTV</th>
<th>FFDNet</th>
<th>DeSCI</th>
<th>FFDNet-TV</th>
<th>Ours (2DTV)</th>
<th>Ours (3DTV)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>30.70, 0.8812</td>
<td>30.15, 0.8906</td>
<td>28.65, 0.8339</td>
<td>31.71, 0.9153</td>
<td>31.26, 0.8867</td>
<td><strong>34.46, 0.9318</strong></td>
<td><strong>34.79, 0.9347</strong></td>
</tr>
</tbody>
</table>
Comparison with Non-DL Methods
## Comparison with DL Methods

### Table 1. Comparison with learned prior method AE [1].

<table>
<thead>
<tr>
<th></th>
<th>103</th>
<th>101</th>
<th>73</th>
<th>92</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>AE</td>
<td>36.75, 0.9726</td>
<td><strong>38.35, 0.9694</strong></td>
<td>38.19, 0.9635</td>
<td>32.49, 0.8874</td>
<td>36.45, <strong>0.9482</strong></td>
</tr>
<tr>
<td>3DVT</td>
<td>34.65, 0.9590</td>
<td>35.96, 0.9396</td>
<td>36.36, 0.9490</td>
<td>31.35, 0.8516</td>
<td>34.58, 0.9248</td>
</tr>
<tr>
<td>DeSCI</td>
<td>23.98, 0.8053</td>
<td>26.15, 0.8334</td>
<td>28.36, 0.8492</td>
<td>20.76, 0.6865</td>
<td>24.81, 0.7936</td>
</tr>
<tr>
<td><strong>Our (3DVT)</strong></td>
<td><strong>37.05, 0.9735</strong></td>
<td>38.14, 0.9599</td>
<td><strong>38.56, 0.9603</strong></td>
<td><strong>32.96, 0.8989</strong></td>
<td><strong>36.68, 0.9482</strong></td>
</tr>
</tbody>
</table>

### Table 2. Comparison with deep learning method $\lambda$-Net.

<table>
<thead>
<tr>
<th></th>
<th>3DVT</th>
<th>2DVT</th>
<th>FFDNet</th>
<th>FFDNet-TV</th>
<th>DeSCI</th>
<th>$\lambda$-Net</th>
<th>Ours (3DVT)</th>
<th>Ours (2DVT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scene 1</td>
<td>31.83, 0.9179</td>
<td>34.68, 0.9321</td>
<td>27.91, 0.8794</td>
<td>35.08, 0.9237</td>
<td>36.07, 0.9505</td>
<td>37.99, 0.8971</td>
<td>37.12, 0.9626</td>
<td>38.16, 0.9630</td>
</tr>
<tr>
<td>Scene 2</td>
<td>22.38, 0.7542</td>
<td>27.39, 0.9165</td>
<td>22.55, 0.7463</td>
<td>27.39, 0.9114</td>
<td>30.64, 0.9580</td>
<td>32.70, 0.9465</td>
<td>30.11, 0.9558</td>
<td>30.11, 0.9566</td>
</tr>
<tr>
<td>Scene 3</td>
<td>28.66, 0.8851</td>
<td>28.52, 0.8888</td>
<td>21.75, 0.7236</td>
<td>28.04, 0.8883</td>
<td>29.87, 0.9132</td>
<td>34.02, 0.9524</td>
<td>31.70, 0.9156</td>
<td>31.38, 0.9040</td>
</tr>
<tr>
<td>Scene 4</td>
<td>25.44, 0.8205</td>
<td>31.72, 0.9309</td>
<td>24.97, 0.8055</td>
<td>32.47, 0.9303</td>
<td>40.35, 0.9780</td>
<td>30.11, 0.9247</td>
<td>31.89, 0.9404</td>
<td>38.25, 0.9691</td>
</tr>
<tr>
<td>Scene 5</td>
<td>31.05, 0.8782</td>
<td>31.63, 0.8599</td>
<td>26.17, 0.7643</td>
<td>32.02, 0.8547</td>
<td>33.86, 0.9038</td>
<td>38.10, 0.9330</td>
<td>34.73, 0.9313</td>
<td>34.70, 0.9265</td>
</tr>
<tr>
<td>Scene 6</td>
<td>26.47, 0.8517</td>
<td>28.04, 0.8613</td>
<td>26.11, 0.8725</td>
<td>28.79, 0.8525</td>
<td>33.59, 0.9421</td>
<td>30.73, 0.9222</td>
<td>31.44, 0.9234</td>
<td>32.82, 0.9326</td>
</tr>
<tr>
<td>Scene 7</td>
<td>29.20, 0.8841</td>
<td>33.90, 0.9287</td>
<td>24.81, 0.8128</td>
<td>34.31, 0.9279</td>
<td>35.76, 0.9515</td>
<td>37.15, 0.9675</td>
<td>34.70, 0.9431</td>
<td>36.17, 0.9476</td>
</tr>
<tr>
<td>Scene 8</td>
<td>26.94, 0.8716</td>
<td>30.12, 0.8779</td>
<td>21.64, 0.7117</td>
<td>30.29, 0.8703</td>
<td>31.34, 0.9061</td>
<td>34.35, 0.9454</td>
<td>30.12, 0.9086</td>
<td>31.62, 0.9044</td>
</tr>
<tr>
<td>Scene 9</td>
<td>33.31, 0.9350</td>
<td>35.31, 0.9569</td>
<td>37.64, 0.9485</td>
<td>35.90, 0.9541</td>
<td>40.87, 0.9694</td>
<td>36.04, 0.9264</td>
<td>37.07, 0.9705</td>
<td>40.56, 0.9746</td>
</tr>
<tr>
<td>Scene 10</td>
<td>25.23, 0.8307</td>
<td>27.59, 0.8431</td>
<td>20.37, 0.6289</td>
<td>27.83, 0.8400</td>
<td>28.96, 0.8746</td>
<td>29.47, 0.9062</td>
<td>28.81, 0.8803</td>
<td>28.99, 0.8731</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td>28.05, 0.8629</td>
<td>30.89, 0.8996</td>
<td>25.39, 0.7893</td>
<td>31.21, 0.8953</td>
<td>34.13, 0.9347</td>
<td>34.07, 0.9321</td>
<td>32.77, 0.9332</td>
<td><strong>34.28, 0.9352</strong></td>
</tr>
</tbody>
</table>
Extension to Video CS

Truth | FFDNet | DnCNN | Ours

sensing ratio: 1/8
Summary

- Snapshot compressive imaging is an effective way to capture HD image data
- Our PnP algorithms are very flexible
- Several interesting problems need to be further investigated: recovery theory, convergence rate, ...

References:
- Haiquan Guo, Yao Wang, Deyu Meng, Effective Snapshot Compressive-spectral Imaging via Deep Denoising and Total Variation Priors, CVPR, 2021
- Shirin Jalali, Xin Yuan, Snapshot Compressed Sensing: Performance Bounds and Algorithms, IEEE TIT, 2019
TV Regularized Nonlocal Low-rank Tensor Train

Learn nonlocal structure from the measurement

Group similar cubes on the learned clusters and employ a low tensor train rank constraint on these 4D tensors

Output

Reconstructed Scene
TV Regularized Nonlocal Low-rank Tensor Train

Truth
PSNR/SSIM
30.89/0.93
5m

GAP-TV
PSNR/SSIM
35.03/0.96
100m

DeSCI
PSNR/SSIM
31.28/0.93
1m

FFDNet
PSNR/SSIM
36.34/0.96
20m

FFDNet+3DTV
PSNR/SSIM
39.91/0.98
60m

Face image: 512*512*31
Thank you!