Single Image Interpolation via Adaptive Non-Local Sparsity-Based Modeling

Yaniv Romano  
The Electrical Engineering Department

Matan Protter  
The Computer Science Department

Michael Elad  
The Computer Science Department

SIAM Conference on IMAGING SCIENCE

May 2014, Hong Kong
The Interpolation Problem

- Given a Low-Resolution (LR) image

\[ y = U_L x \]

- Our goal is to recover \( x \) from \( y \).

- Decimates the image by a factor of \( L \) along the horizontal and vertical dimensions (without blurring)

- Low-Resolution (LR) image
- High-Resolution (HR) image
Background
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:

\[
\mathbf{x} = D\alpha
\]

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

The computation of \( \alpha \) from \( x \) (or its noisy/decimated versions) is called sparse-coding.

The OMP is a popular sparse-coding technique, especially for low dimensional signals.
The above is one in the large family of patch-based algorithms.

Effective for denoising, deblurring, inpainting, tomographic reconstruction, etc.
Interpolation – Starting Point

- Drawbacks...
  - Each patch is interpolated independently.
  - Sparse-coding tends to err due to small number of existing pixels.

*sparse-coding using high/low weights.
The Basic Idea

- The more known pixels within a patch, the better the restoration:
  - The number of known pixels depends on its location.
  - "strong" patches.
  - "weak" patches.

- We suggest "increasing" the number of known pixels based on the "self-similarity" assumption.
Past Work

- **LSSC** – A non local sparse model for image restoration  
  [Mairal, Bach, Ponce, Sapiro and Zisserman (’09)].
  - Applies joint sparse coding (Simultaneous OMP) instead of OMP.

- **NARM** – Combines the sparsity prior and non-local autoregressive modeling  
  [Dong, Zhang, Lukac and Shi (’13)].
  - Embeds the autoregressive model (i.e. connecting a missing pixel with its nonlocal neighbors) into the conventional sparsity data fidelity term.

- **PLE** – A framework for solving inverse problems based on Gaussian mixture model  
  [Yu, Sapiro and Mallat (’12)].
  - Very effective for denoising, inpainting, interpolation and deblurring.
The Algorithm
Outline

LR Input image → Basic interpolated image

→ Group similar patches together

→ Interpolate each group jointly

→ Reconstruct the image

→ Update the Dictionary
- Combine the "self-similarity" assumption and the sparsity prior.
Sparse-Coding

- Combine the “self-similarity” assumption and the sparsity prior.
  - Use Simultaneous OMP (SOMP) instead of OMP [J. A. Tropp et al. (’06)].
Combine the "self-similarity" assumption and the sparsity prior.

- Use Simultaneous OMP (SOMP) instead of OMP [J. A. Tropp et al. ('06)].
  - Improve the sparse-coding by finding the joint representation of a non-weighted version of the reference patch ("stabilizer") and its K-NN.
Combine the “self-similarity” assumption and the sparsity prior.

- Use Simultaneous OMP (SOMP) instead of OMP [J. A. Tropp et al. (’06)].
  - Improve the sparse-coding by finding the joint representation of a non-weighted version of the reference patch (“stabilizer”) and its K-NN.

Given the representations, we update the dictionary.

- Using a weighted version of the KSVD.
Combining the “self-similarity” assumption and the sparsity prior.

- Use Simultaneous OMP (SOMP) instead of OMP [J. A. Tropp et al. (’06)].
  - Improve the sparse-coding by finding the joint representation of a non-weighted version of the reference patch (“stabilizer”) and its K-NN.

Given the representations, we update the dictionary.

- Using a weighted version of the KSVD.

Reconstruct the HR image.

- How?
A two-stage algorithm:

1. **First stage** exploits “strong” patches.

- **Per each patch:** Find its K-Nearest “strong” Neighbors
- **Interpolate** each group jointly (using the “stabilizer”)
- **Reconstruct** the image by exploiting the “strong” patches
- **Initial cubic HR estimation**
- **Update the Dictionary**
A two-stage algorithm:

1. **First stage** exploits “strong” patches.
2. **Second stage** obtains the final HR image.

**Per each patch:**
- Find its K-Nearest “strong” Neighbors
- **Interpolate** each group jointly (using the “stabilizer”)
- **Reconstruct** the image by exploiting the “strong” patches

**First stage’ HR image**
**First stage’ Dictionary**
The Proposed Method

- A two-stage algorithm:
  1. **First stage** exploits “strong” patches.
  2. **Second stage** obtains the final HR image.

The "general" method

The proposed two-stage method
Experiments
We test the proposed algorithm over 18 well-known images.

Each image was decimated (without blurring) by a factor of 2 or 3 along each axis.

We compared the PSNR* of the proposed algorithm to the current state-of-the-art methods.

*PSNR = $20\log_{10} \left( \frac{255}{\sqrt{\text{MSE}}} \right)$
## Quantitative Results [dB]

Average PSNR* over 18 well-known images (higher is better):

<table>
<thead>
<tr>
<th>Method</th>
<th>Scaling Factor</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X 2</td>
<td>X 3</td>
<td></td>
</tr>
<tr>
<td>Cubic</td>
<td>28.98</td>
<td>25.52</td>
<td></td>
</tr>
<tr>
<td>SAI [Zhang &amp; Wu ('08)]</td>
<td>29.51</td>
<td>25.83</td>
<td></td>
</tr>
<tr>
<td>SME [Mallat &amp; Yu ('10)]</td>
<td>29.62</td>
<td>25.95</td>
<td></td>
</tr>
<tr>
<td>PLE [Yu, Sapiro &amp; Mallat ('12)]</td>
<td>29.62</td>
<td>26.08</td>
<td></td>
</tr>
<tr>
<td>NARM [Dong et al. ('13)]</td>
<td><strong>29.98</strong></td>
<td><strong>26.21</strong></td>
<td></td>
</tr>
</tbody>
</table>

*PSNR = 20\log_{10}\left(\frac{255}{\sqrt{MSE}}\right)
Quantitative Results [dB]

- Average PSNR* over 18 well-known images (higher is better):

<table>
<thead>
<tr>
<th>Method</th>
<th>Scaling Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X 2</td>
</tr>
<tr>
<td>Cubic</td>
<td>28.98</td>
</tr>
<tr>
<td>SAI [Zhang &amp; Wu ('08)]</td>
<td>29.51</td>
</tr>
<tr>
<td>SME [Mallat &amp; Yu ('10)]</td>
<td>29.62</td>
</tr>
<tr>
<td>PLE [Yu, Sapiro &amp; Mallat ('12)]</td>
<td>29.62</td>
</tr>
<tr>
<td>NARM [Dong et al. ('13)]</td>
<td>29.98</td>
</tr>
<tr>
<td>Proposed</td>
<td><strong>30.09</strong></td>
</tr>
<tr>
<td>Diff to best result</td>
<td>0.11</td>
</tr>
</tbody>
</table>

*PSNR = 20\log_{10} \left( \frac{255}{\sqrt{MSE}} \right)
Visual Comparison (X2)

Original  SAI  SME

PLE  NARM  Proposed
Visual Comparison (X2)

Original
Visual Comparison (X2)

NARM
Visual Comparison (X2)

Proposed
Visual Comparison (X2)

- Original
- SAI
- SME
- PLE
- NARM
- Proposed
Visual Comparison (X2)
Visual Comparison (X2)
Visual Comparison (X2)

Original  
SAI  
SME  

PLE  
NARM  
Proposed  

NARM
Visual Comparison (X2)

Original  SAI  SME

PLE  NARM  Proposed
Visual Comparison (X3)

Original  SAI  SME

PLE  NARM  Proposed

Original  PLE

SAI  NARM

SME  Proposed
Visual Comparison (X3)
Visual Comparison (**X3**)
Visual Comparison (X3)

Original  SAI  SME

PLE  NARM  Proposed

Proposed
Visual Comparison (X3)
Visual Comparison (X3)

Original  SAI  SME

PLE  NARM  Proposed

Original
Visual Comparison (X3)

Original  SAI  SME  NARM

PLE  NARM  Proposed
Visual Comparison (X3)

Original  SAI  SME

PLE  NARM  Proposed

Proposed
Conclusions

A decimated HR image in a regular pattern

Image interpolation by K-SVD inpainting isn’t competitive

We shall use our two-stage algorithm that combines the non-local proximity and sparsity priors

The 1\textsuperscript{st} stage trains a dictionary and obtains a HR image by exploiting the “strong” patches both in the sparse-coding and reconstruction steps

The 2\textsuperscript{nd} stage builds upon the 1\textsuperscript{st} stage results and obtains the final image

This leads to state-of-the-art results
We Are Done

Thank You

Questions ?