Sparse Coding

Acknowledgement: The slides are adapted from the ones prepared by M. Elad.

Denoising By Energy Minimization

Many of the proposed image denoising algorithms are related to the minimization of an energy function of the form

\[ f(x) = \frac{1}{2} \| x - y \|_2^2 + G(x) \]

- \(y\) : Given measurements
- \(x\) : Unknown to be recovered

- This is in-fact a Bayesian point of view, adopting the Maximum-A-posteriori Probability (MAP) estimation.
- Clearly, the wisdom in such an approach is within the choice of the prior – modeling the images of interest.

The Evolution of G(x)

During the past several decades we have made all sort of guesses about the prior \(G(x)\) for images:

- \(G(x) = \lambda \| x \|_2^2\) (Energy)
- \(G(x) = \lambda \| L x \|_2^2\) (Smoothness)
- \(G(x) = \lambda \| L x \|_W^2\) (Adapt+Smooth)
- \(G(x) = \lambda \{ L x \}\) (Robust Statistics)

- \(G(x) = \lambda \| \nabla x \|_2\) (Total-Variation)
- \(G(x) = \lambda \| W x \|_2\) (Wavelet Sparsity)
- \(G(x) = \lambda \| D x \|_2\) for \(x = D u\) (Sparse & Redundant)

- Hidden Markov Models,
- Compression algorithms as priors,
- …
Sparse Modeling of Signals

- Every column in $\mathbf{D}$ (dictionary) is a prototype signal (atom).
- The vector $\alpha$ is generated randomly with few (say $L$) non-zeros at random locations and with random values.
- We shall refer to this model as Sparseland

Interesting Model:
- **Simple**: Every generated signal is built as a linear combination of few atoms from our dictionary $\mathbf{D}$
- **Rich**: A general model: the obtained signals are a union of many low-dimensional Gaussians.
- **Familiar**: We have been using this model in other context for a while now (wavelet, JPEG, …).

Sparse & Redundant Rep. Modeling?

Our signal model is thus: $x = \mathbf{D} \alpha$ where $\alpha$ is sparse

Sparse & Redundant Rep. Modeling?

$$\|\alpha\|_p^p = \sum_{j=1}^{k} |\alpha_j|^p$$

Our signal model is thus: $x = \mathbf{D} \alpha$ where $\alpha$ is sparse
Sparse & Redundant Rep. Modeling?

As \( p \to 0 \) we get a count of the non-zeros in the vector:

\[
\|\alpha\|_0^0
\]

Our signal model is thus:

\[
x = D\alpha \quad \text{where} \quad \|\alpha\|_0^0 \leq L
\]

Back to Our MAP Energy Function

- \( L_0 \) norm effectively counts the number of non-zeros in \( \alpha \).
- The vector \( \alpha \) is the representation (sparse/redundant) of the desired signal \( x \).
- The core idea: while few (L out of K) atoms can be merged to form the true signal, the noise cannot be fitted well. Thus, we obtain an effective projection of the noise onto a very low-dimensional space, thus getting denoising effect.

Wait! There are Some Issues

- **Numerical Problems:** How should we solve or approximate the solution of the problem:

\[
\min_{\alpha} \|D\alpha - y\|_2^2 \quad \text{s.t.} \quad \|\alpha\|_0^0 \leq L \quad \text{or} \quad \min_{\alpha} \|\alpha\|_0^0 \quad \text{s.t.} \quad \|D\alpha - y\|_2^2 \leq \varepsilon^2
\]

or

\[
\min_{\alpha} \lambda \|\alpha\|_1 + \|D\alpha - y\|_2^2
\]

- **Theoretical Problems:** Is there a unique sparse representation? If we are to approximate the solution somehow, how close will we get?

- **Practical Problems:** What dictionary \( D \) should we use, such that all this leads to effective denoising? Will all this work in applications?

To Summarize So Far ...

- Image denoising (and many other problems in image processing) requires a model for the desired image.

- What can we do?

- Use a model for signals/images based on sparse and redundant representations.

- There are some issues:
  1. Theoretical
  2. How to approximate?
  3. What about \( D \)?

Great! No!
**Let's Start with the Noiseless Problem**

Suppose we build a signal by the relation

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

We aim to find the signal's representation:

\[ \hat{\alpha} = \text{ArgMin}_{\alpha} \| \alpha \|^0 \quad \text{s.t.} \quad \mathbf{x} = \mathbf{D} \alpha \]

Why should we necessarily get \( \hat{\alpha} = \alpha \)?

It might happen that eventually \( \| \hat{\alpha} \|^0 < \| \alpha \|^0 \).

**Uniqueness**

**Uniqueness Rule**

Suppose this problem has been solved somehow

\[ \hat{\alpha} = \text{ArgMin}_{\alpha} \| \alpha \|^0 \quad \text{s.t.} \quad \mathbf{x} = \mathbf{D} \alpha \]

If we found a representation that satisfy

\[ \| \hat{\alpha} \|^0 < \frac{\sigma}{2} \]

Then necessarily it is unique (the sparsest).

This result implies that if \( \mathcal{M} \) generates signals using "sparse enough" \( \alpha \), the solution of the above will find it exactly.

**Matrix “Spark”**

Definition: Given a matrix \( \mathbf{D} \), \( \sigma = \text{Spark}(\mathbf{D}) \) is the smallest number of columns that are linearly dependent.

*Donoho & E. ('02)*

Example:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0
\end{bmatrix}
\]

- **Rank** = 4
- **Spark** = 3

*In tensor decomposition, Kruskal defined something similar already in 1989.*

**Our Goal**

\[
\min_{\alpha} \| \alpha \|^0 \quad \text{s.t.} \quad \| \mathbf{D} \alpha - \mathbf{y} \|^2 \leq \varepsilon^2
\]

This is a combinatorial problem, proven to be NP-Hard!

Here is a recipe for solving this problem:

1. **Set** \( L = 1 \)
2. Gather all the supports \( \{s_i\} \) of cardinality \( L \)
3. For each support, solve the LS problem
   \[
   \min_{\alpha} \quad \| \mathbf{D} \alpha - \mathbf{y} \|^2 \leq \varepsilon^2
   \]
   \( \text{supp}(\alpha) = S \)
4. If \( \text{LS error} \leq \varepsilon^2 \)
   - **Yes**
   - **Done**
5. **Else**
   - **No**
   - **Set** \( L = L + 1 \)

Assume: \( K = 1000, L = 10 \) (known!), 1 nano-sec per each LS

We shall need \( \sim 8 \times 10^6 \) years to solve this problem !!!!!
**Lets Approximate**

\[
\min_{\alpha} \|\alpha\|_0 \quad \text{s.t.} \quad \|D\alpha - y\|_2^2 \leq \epsilon^2
\]

**Relaxation methods**
Smooth the \(L_0\) and use continuous optimization techniques

**Greedy methods**
Build the solution one non-zero element at a time

**Go Greedy: Matching Pursuit (MP)**

- The MP is one of the greedy algorithms that finds one atom at a time [Mallat & Zhang (93)].
- Step 1: find the one atom that best matches the signal.
- Next steps: given the previously found atoms, find the next one to best fit the residual.
- The algorithm stops when the error \(\|D\alpha - y\|_2\) is below the destination threshold.
- The Orthogonal MP (OMP) is an improved version that re-evaluates the coefficients by Least-Squares after each round.

**Relaxation – The Basis Pursuit (BP)**

Instead of solving

\[
\min_{\alpha} \|\alpha\|_0 \quad \text{s.t.} \quad \|D\alpha - y\|_2 \leq \epsilon
\]

Solve Instead

\[
\min_{\alpha} \|\alpha\|_1 \quad \text{s.t.} \quad \|D\alpha - y\|_2 \leq \epsilon
\]

- This is known as the Basis-Pursuit (BP) [Chen, Donoho & Saunders (95)].
- The newly defined problem is convex (quad. programming).
- Very efficient solvers can be deployed:
  - Interior point methods [Chen, Donoho, & Saunders (95)] [Kim, Koh, Lustig, Boyd, & D. Gorinevsky (07)].
  - Sequential shrinkage for union of ortho-bases [Bruce et al. (98)].
  - Iterative shrinkage [Figuerido & Nowak (03)] [Daubechies, Defrise, & De-Mole (04)] [E. (05)] [E., Matalon, & Zibulevsky (06)] [Beck & Teboulle (09)]

**Pursuit Algorithms**

\[
\min_{\alpha} \|\alpha\|_0 \quad \text{s.t.} \quad \|D\alpha - y\|_2^2 \leq \epsilon^2
\]

There are various algorithms designed for approximating the solution of this problem:

- Relaxation Algorithms: Basis Pursuit (a.k.a. LASSO), Dnatzig Selector & numerical ways to handle them [1995-today].
- …
Pursuit Algorithms

There are various algorithms designed for approximating the solution of this problem:

- Relaxation Algorithms: Basis Pursuit (a.k.a. LASSO), Dantzig Selector & numerical ways to handle them [1995-today].

Why should they work?

The Mutual Coherence

The Mutual Coherence $\mu$ is the largest off-diagonal entry in absolute value.

The Mutual Coherence is a property of the dictionary (just like the “Spark”). In fact, the following relation can be shown:

$$\sigma \geq 1 + \frac{1}{\mu}$$

BP and MP Equivalence (No Noise)

Given a signal $x$ with a representation $x = D\alpha$, assuming that $\|x\|_0 < 0.5(1 + 1/\mu)$, BP and MP are guaranteed to find the sparsest solution.

- MP and BP are different in general (hard to say which is better).
- The above result corresponds to the worst-case, and as such, it is too pessimistic.
- Average performance results are available too, showing much better bounds [Donoho (04)] [Candes et al. (04)] [Tanner et al. (05)] [E (06)] [Tropp et al. (06)] … [Candes et al. (09)].
BP Stability for the Noisy Case

\[ \min_{\alpha} \lambda \|\alpha\|_1 + \|D\alpha - y\|_2^2 \]

To Summarize So Far …

Image denoising (and many other problems in image processing) requires a model for the desired image.

What do we do?

Use a model for signals/images based on sparse and redundant representations.

Problems?

The Dictionary \( D \) should be found somehow !!!

What next?

We have seen that there are approximation methods to find the sparsest solution, and there are theoretical results that guarantee their success.

What Should \( D \) Be?

Our Assumption: Good-behaved Images have a sparse representation.

One approach to choose \( D \) is from a known set of transforms (Steerable wavelet, Curvelet, Contourlets, Bandlets, Shearlets …)

The approach we will take for building \( D \) is training it, based on Learning from Image Examples.

To Build \( D \):

1. Use a model for signals/images based on sparse and redundant representations.
2. Find the Dictionary \( D \) somehow.
3. Use approximation methods to find the sparsest solution.
4. Guarantee their success with theoretical results.

What do we get?

• For \( \sigma = 0 \), we get a weaker version of the previous result.
• This result is the oracle’s error, multiplied by \( C \cdot \log K \).
• Similar results exist for other pursuit algorithms (Dantzig Selector, Orthogonal Matching Pursuit, CoSaMP, Subspace Pursuit, …).

Ben-Haim, Eldar & E. ('09)

Stability

Given a signal \( y = D\alpha + \nu \) with a representation satisfying \( \|\alpha\|_0 < 1 / 3\mu \) and a white Gaussian noise \( \nu \sim N(0, \sigma^2I) \), BP will show stability, i.e.,

\[ \|\hat{x}_{\text{BP}} - \alpha\|_2^2 < \text{Const}(\lambda) \cdot \log K \cdot \|\alpha\|_0 \cdot \sigma^2 \]

*With very high probability
Each example is a linear combination of atoms from D. Measure of Quality for D:

\[
\min_{D,A} \sum_{j=1}^{P} \|D\alpha_j - x_j\|_2^2 \quad \text{s.t.} \quad \forall j, \|\alpha_j\|_0 \leq L
\]

Each example has a sparse representation with no more than L atoms.

K–Means For Clustering:

Clustering: An extreme sparse representation

**Initialize** D

**Sparse Coding**

Nearest Neighbor

**Dictionary Update**

Column-by-Column by Mean computation over the relevant examples

The K–SVD Algorithm – General

[Aharon, E. & Bruckstein (’04, ’05)]

**Initialize** D

**Sparse Coding**

Use Matching Pursuit

**Dictionary Update**

Column-by-Column by SVD computation over the relevant examples

K–SVD: Sparse Coding Stage

\[
\min_{\alpha} \sum_{j=1}^{P} \|D\alpha_j - x_j\|_2^2 \quad \text{s.t.} \quad \forall j, \|\alpha_j\|_p \leq L
\]

D is known! For the jth item we solve

\[
\min_{\alpha} \|D\alpha_j - x_j\|_2^2 \quad \text{s.t.} \quad \|\alpha_j\|_p \leq L
\]

Solved by A Pursuit Algorithm

[Field & Olshausen ('96)]
[Engan et al. ('99)]
[Lewicki & Sejnowski ('99)]
[Coates et al. ('03)]
[Gribonval et al. ('04)]
[Aharon, E. & Bruckstein ('04)]
[Aharon, E. & Bruckstein ('05)]
[Lewicki & Sejnowski ('00)]
[Cotter et al. ('03)]
[Gribonval et al. ('04)]
[Aharon, E. & Bruckstein ('04)]
[Aharon, E. & Bruckstein ('05)]
K–SVD: Dictionary Update Stage

Refer only to the examples that use the column $d_k$.

We should solve:

$$\begin{align*}
\text{Min} & \quad \| \alpha_k d_k^T - E \|_F^2 \\
\text{subject to} & \quad \alpha_k, d_k
\end{align*}$$

Fixing all $A$ and $D$ apart from the $k^{th}$ column, and seek both $d_k$ and the $k^{th}$ column in $A$ to better fit the residual!

To Summarize So Far …

Use a model for signals/images based on sparse and redundant representations

We have seen that there are approximation methods to find the sparsest solution, and there are theoretical results that guarantee their success.

• The K-SVD algorithm is reasonable for low-dimension signals ($N$ in the range 10-400). As $N$ grows, the complexity and the memory requirements of the K-SVD become prohibitive.

• So, how should large images be handled?

From Local to Global Treatment

• The solution: Force shift-invariant sparsity - on each patch of size $N$-by-$N$ ($N=8$) in the image, including overlaps.

$$\hat{x} = \text{ArgMin}_{\chi, \{g_{ij}\}_{ij}} \frac{1}{2} \| x - y \|_2^2 + \mu \sum_{ij} \| R_{ij} x - D g_{ij} \|_2^2$$

s.t. $\| g_{ij} \|_0 \leq L$

Extracts a patch in the $ij$ location

Our prior
**What Data to Train On?**

**Option 1:**
- Use a database of images,
- We tried that, and it works fine (~0.5-1dB below the state-of-the-art).

**Option 2:**
- Use the corrupted image itself!!
- Simply sweep through all patches of size \( N \times N \) (overlapping blocks),
- Image of size \( 1000^2 \) pixels \( \approx 10^6 \) examples to use – more than enough.
- This works much better!

---

**K-SVD Image Denoising**

\[
\hat{x} = \text{ArgMin} \left\{ \frac{1}{2} \|D\hat{x} - y\|^2_F + \frac{\mu}{2} \sum_{ij} |R_{ij}| x_D - D_{ij} \|_F^2 \right\}, \quad \text{s.t. } \|D_{ij}\|^2_F \leq L
\]

Compute \( \alpha_{ij} \) per patch \( \alpha_{ij} = \min_{\alpha} |R_{ij}| \hat{x} - D_{ij}\|^2_F \)

Compute \( D \) to minimize \( \min_{\alpha} \sum_{ij} |R_{ij}| x_D - D_{ij}\|^2_F \)

using the matching pursuit

Compute \( x \) by \( x = \left[ \text{arg} \left\{ \sum_{ij} |R_{ij}| D_{ij}\right\} \right] \)

which is a simple averaging of shifted patches

---

**Image Denoising (Gray)** [E. & Aharon ('06)]

- **Source**
- **Noisy image** \( \sigma = 20 \)
- **Result**: 30.829dB
- **The obtained dictionary after 10 iterations**

---

**Image Denoising (Gray)** [E. & Aharon ('06)]

- **Source**
- **Noisy image** \( \sigma = 20 \)
- **Result**: 30.829dB
- **The obtained dictionary after 10 iterations**

---

- The results of the K-SVD algorithm compete favorably with the state-of-the-art.
- In a recent work that extended this algorithm to use joint sparse representation on the patches, the best published denoising performance are obtained [Mairal, Bach, Ponce, Sapiro & Zisserman ('09)].
Denoising (Color) [Mairal, E. & Sapiro ('08)]

- When turning to handle color images, the main difficulty is in defining the relation between the color layers – R, G, and B.
- The solution with the above algorithm is simple – consider 3D patches or 8-by-8 with the 3 color layers, and the dictionary will detect the proper relations.

The K-SVD algorithm leads to state-of-the-art denoising results, giving ~1dB better results compared to [Mcauley et al. ('06)] which implements a learned MRF model (Field-of-Experts).

Original            Noisy (12.77dB)     Result  (29.87dB)

Video Denoising [Protter & E. ('09)]

When turning to handle video, one could improve over the previous scheme in three important ways:

1. Propagate the dictionary from one frame to another, and thus reduce the number of iterations;
2. Use 3D patches that handle the motion implicitly; and
3. Motion estimation and compensation can and should be avoided [Buades, Col, and Morel ('06)].
**Video Denoising [Protter & E. ('09)]**

The K-SVD algorithm leads to state-of-the-art video denoising results, giving ~0.5dB better results on average compared to [Boades, Coll & Morel ('05)] and comparable to [Rusanovskyy, Dabov, & Egiazarian ('06)].

**Low-Dosage Tomography [Shtok, Zibulevsky & E. ('10)]**

- In Computer-Tomography (CT) reconstruction, an image is recovered from a set of its projections.
- In medicine, CT projections are obtained by X-ray, and it typically requires a high dosage of radiation in order to obtain a good quality reconstruction.
- A lower-dosage projection implies a stronger noise (Poisson distributed) in data to work with.
- Armed with sparse and redundant representation modeling, we can denoise the data and the final reconstruction ... enabling CT with lower dosage.

**Low-Dosage Tomography [Shtok, Zibulevsky & E. ('10)]**

- FBP result with high dosage
  - PSNR=24.63dB
- FBP result with low dosage (one fifth)
  - PSNR=22.31dB
- Denoising of the sinogram and post-processing (another denoising stage) of the reconstruction
  - PSNR=26.06dB
**Image Inpainting – The Basics**

- Assume: the signal $\mathbf{x}$ has been created by $\mathbf{x} = \mathbf{D} \mathbf{\alpha}_0$ with very sparse $\mathbf{\alpha}_0$.
- Missing values in $\mathbf{x}$ imply missing rows in this linear system.
- By removing these rows, we get $\tilde{\mathbf{D}} \mathbf{\alpha} = \tilde{\mathbf{x}}$.
- Now solve
  $$\min_{\mathbf{\alpha}} \|\mathbf{\alpha}\|_0 \quad \text{s.t.} \quad \tilde{\mathbf{x}} = \tilde{\mathbf{D}} \mathbf{\alpha}$$
- If $\mathbf{\alpha}_0$ was sparse enough, it will be the solution of the above problem! Thus, computing $\mathbf{D} \mathbf{\alpha}_0$ recovers $\mathbf{x}$ perfectly.

**Side Note: Compressed-Sensing**

- Compressed Sensing is leaning on the very same principal, leading to alternative sampling theorems.
- Assume: the signal $\mathbf{x}$ has been created by $\mathbf{x} = \mathbf{D} \mathbf{\alpha}_0$ with very sparse $\mathbf{\alpha}_0$.
- Multiply this set of equations by the matrix $\mathbf{Q}$ which reduces the number of rows.
- The new, smaller, system of equations is
  $$\mathbf{Q} \mathbf{D} \mathbf{\alpha} = \mathbf{Q} \mathbf{x} \quad \Rightarrow \quad \tilde{\mathbf{D}} \mathbf{\alpha} = \mathbf{Q} \mathbf{x} \times \mathbf{\mathbf{D}}$$
- If $\mathbf{\alpha}_0$ was sparse enough, it will be the sparsest solution of the new system, thus, computing $\mathbf{D} \mathbf{\alpha}_0$ recovers $\mathbf{x}$ perfectly.
- Compressed sensing focuses on conditions for this to happen, guaranteeing such recovery.

**Inpainting** [Mairal, E. & Sapiro ('08)]

Experiments lead to state-of-the-art inpainting results.

Original | 80% missing | Result
---|---|---

**Inpainting** [Mairal, E. & Sapiro ('08)]

Experiments lead to state-of-the-art inpainting results.

Original | 80% missing | Result
---|---|---
Experiments lead to state-of-the-art inpainting results.

The same can be done for video, very much like the denoising treatment: (i) 3D patches, (ii) no need to compute the dictionary from scratch for each frame, and (iii) no need for explicit motion estimation.

Demosaicing [Mairal, E. & Sapiro ('08)]

- Today’s cameras are sensing only one color per pixel, leaving the rest for interpolated.
- Generalizing the inpainting scheme to handle demosaicing is tricky because of the possibility to learn the mosaic pattern within the dictionary.
- In order to avoid “over-fitting”, we handle the demosaicing problem while forcing strong sparsity and applying only few iterations.

Experiments lead to state-of-the-art demosaicing results, giving ~0.2dB better results on average, compared to [Chang & Chan ('06)].
**Image Compression** [Bryt and E. (’08)]

- The problem: Compressing photo-ID images.
- *General* purpose methods (JPEG, JPEG2000) do not take into account the specific family.
- By *adapting* to the image-content (PCA/K-SVD), better results could be obtained.
- For these techniques to operate well, *train dictionaries locally* (per patch) using a training set of images is required.
- In PCA, only the (quantized) coefficients are stored, whereas the K-SVD requires storage of the indices well.
- *Geometric* alignment of the image is very helpful and should be done [Goldenberg, Kimmel, & E. (’05)].

---

**Image Compression**

- Detect main features and warp the images to a common reference (20 parameters)
- Divide the image into disjoint 15-by-15 patches. For each compute mean and dictionary
- Per each patch find the operating parameters (number of atoms L, quantization Q)
- Warp, remove the mean from each patch, sparse code using L atoms, apply Q, and dewarp

---

**Image Compression Results**

<table>
<thead>
<tr>
<th></th>
<th>Original</th>
<th>JPEG</th>
<th>JPEG-2000</th>
<th>Local-PCA</th>
<th>K-SVD</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>820 Bytes</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Results</td>
<td>11.99</td>
<td>10.40</td>
<td>8.81</td>
<td>5.56</td>
<td></td>
</tr>
<tr>
<td></td>
<td>11.99</td>
<td>10.40</td>
<td>8.81</td>
<td>5.56</td>
<td></td>
</tr>
<tr>
<td></td>
<td>11.99</td>
<td>10.40</td>
<td>8.81</td>
<td>5.56</td>
<td></td>
</tr>
<tr>
<td></td>
<td>11.99</td>
<td>10.40</td>
<td>8.81</td>
<td>5.56</td>
<td></td>
</tr>
</tbody>
</table>

---

**Image Compression Results**

<table>
<thead>
<tr>
<th></th>
<th>Original</th>
<th>JPEG</th>
<th>JPEG-2000</th>
<th>Local-PCA</th>
<th>K-SVD</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>550 Bytes</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Results</td>
<td>15.58</td>
<td>13.99</td>
<td>12.57</td>
<td>10.66</td>
<td>6.60</td>
</tr>
<tr>
<td></td>
<td>15.58</td>
<td>13.99</td>
<td>12.57</td>
<td>10.66</td>
<td>6.60</td>
</tr>
<tr>
<td></td>
<td>15.58</td>
<td>13.99</td>
<td>12.57</td>
<td>10.66</td>
<td>6.60</td>
</tr>
<tr>
<td></td>
<td>15.58</td>
<td>13.99</td>
<td>12.57</td>
<td>10.66</td>
<td>6.60</td>
</tr>
</tbody>
</table>
Image Compression Results

<table>
<thead>
<tr>
<th>Original</th>
<th>JPEG</th>
<th>JPEG-2000</th>
<th>Local-PCA</th>
<th>K-SVD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Results for 400 Bytes per each file</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
</tbody>
</table>

Deblocking the Results [Bryt and E. (09)]

<table>
<thead>
<tr>
<th>K-SVD</th>
<th>K-SVD</th>
<th>K-SVD</th>
<th>K-SVD</th>
</tr>
</thead>
<tbody>
<tr>
<td>18.62</td>
<td>16.12</td>
<td>16.81</td>
<td>12.30</td>
</tr>
<tr>
<td>11.38</td>
<td>11.57</td>
<td>12.54</td>
<td>7.61</td>
</tr>
</tbody>
</table>

Super-Resolution [Zeyde, Protter, & E. (11)]

- Given a low-resolution image, we desire to enlarge it while producing a sharp looking result. This problem is referred to as “Single-Image Super-Resolution”.
- Image scale-up using bicubic interpolation is far from being satisfactory for this task.
- Recently, a sparse and redundant representation technique was proposed [Yang, Wright, Huang, and Ma (08)] for solving this problem, by training a coupled-dictionaries for the low- and high res. images.
- We extended and improved their algorithms and results.

Super-Resolution – Results (1)

The training image: 717×717 pixels, providing a set of 54,289 training patch-pairs.
Super-Resolution – Results (1)

Ideal Image

SR Result
PSNR=16.95dB

Given Image

Bicubic interpolation
PSNR=14.68dB

Super-Resolution – Results (2)

Given image

Scaled-Up (factor 2:1) using the proposed algorithm.
PSNR=29.32dB (3.32dB improvement over bicubic)

Super-Resolution – Results (2)

The Original

Bicubic Interpolation

SR result