Patch-based regularization in multiband imaging inverse problems

José M. Bioucas Dias

Instituto de Telecomunicações and Instituto Superior Técnico,
Universidade de Lisboa, Portugal

Colllaborators: Mário Figueredo, Afonso Teodoro, Lina Zhuang, Milad Niknejad

TéSA, Telecommunications for Space and Aeronautics, 2018

Distinguished Lecturer Program - GRSS | IEEE

FCT, project UID/EEA/50008/2013

Outline

Outline

- Ill-posed/Ill-Conditioned Inverse Problems

Outline

- Ill-posed/Ill-Conditioned Inverse Problems
- Patch-Based Image regularization

Outline

- Ill-posed/Ill-Conditioned Inverse Problems
- Patch-Based Image regularization
- GMM priors and MMSE denoising

Outline

- Ill-posed/Ill-Conditioned Inverse Problems
- Patch-Based Image regularization
- GMM priors and MMSE denoising
- Plug-and-Play

Outline

- Ill-posed/Ill-Conditioned Inverse Problems
- Patch-Based Image regularization
- GMM priors and MMSE denoising
- Plug-and-Play
- Hyperspectral Image Denoising and Inpainting

Outline

- Ill-posed/Ill-Conditioned Inverse Problems
- Patch-Based Image regularization
- GMM priors and MMSE denoising
- Plug-and-Play
- Hyperspectral Image Denoising and Inpainting
- Hyperspectral Fusion

Outline

- Ill-posed/Ill-Conditioned Inverse Problems
- Patch-Based Image regularization
- GMM priors and MMSE denoising
- Plug-and-Play
- Hyperspectral Image Denoising and Inpainting
- Hyperspectral Fusion
- Final Remarks

Hyperspectral/multispectral image fusion

Hyperspectral/multispectral image fusion

$\mathbf{y}_h = \mathbf{A}_h \mathbf{x} + \mathbf{n}_h$ $\quad$ $\mathbf{y}_p = \mathbf{A}_p \mathbf{x} + \mathbf{n}_p$

$\mathbf{x} \in \mathbb{R}^{600 \times 400 \times 100}$ $\quad$ $\mathbf{y}_h \in \mathbb{R}^{150 \times 100 \times 100}$ $\quad$ $\mathbf{y}_p \in \mathbb{R}^{600 \times 400 \times 1}$

original HS

spatially blurred and downsampled HS

spectrally blurred HS

Multispectral multiresolution imaging (Sentinel 2)

Sentinel-2 Bands	Central Wavelength (µm)	Resolution (m)
Band 1 – Coastal aerosol	0.443	60
Band 2 – Blue	0.490	10
Band 3 – Green	0.560	10
Band 4 – Red	0.665	10
Band 5 – Vegetation Red Edge	0.705	20
Band 6 – Vegetation Red Edge	0.740	20

Sentinel-2 Bands	Central Wavelength (µm)	Resolution (m)
Band 7 – Vegetation Red Edge	0.783	20
Band 8 – NIR	0.842	10
Band 8A – Narrow NIR	0.865	20
Band 9 – Water vapour	0.945	60
Band 10 – SWIR – Cirrus	1.375	60
Band 11 – SWIR	1.610	20
Band 12 – SWIR	2.190	20

Multispectral multiresolution imaging (Sentinel 2)

Sentinel-2 Bands	Central Wavelength (µm)	Resolution (m)
Band 1 – Coastal aerosol	0.443	60
Band 2 – Blue	0.490	10
Band 3 – Green	0.560	10
Band 4 – Red	0.665	10
Band 5 – Vegetation Red Edge	0.705	20
Band 6 – Vegetation Red Edge	0.740	20

Sentinel-2 Bands	Central Wavelength (µm)	Resolution (m)
Band 7 – Vegetation Red Edge	0.783	20
Band 8 – NIR	0.842	10
Band 8A – Narrow NIR	0.865	20
Band 9 – Water vapour	0.945	60
Band 10 – SWIR – Cirrus	1.375	60
Band 11 – SWIR	1.610	20
Band 12 – SWIR	2.190	20

(B4, B3, B2 – 10 m)

Multispectral multiresolution imaging (Sentinel 2)

Sentinel-2 Bands	Central Wavelength (µm)	Resolution (m)
Band 1 – Coastal aerosol	0.443	60
Band 2 – Blue	0.490	10
Band 3 – Green	0.560	10
Band 4 – Red	0.665	10
Band 5 – Vegetation Red Edge	0.705	20
Band 6 – Vegetation Red Edge	0.740	20

Sentinel-2 Bands	Central Wavelength (µm)	Resolution (m)
Band 7 – Vegetation Red Edge	0.783	20
Band 8 – NIR	0.842	10
Band 8A – Narrow NIR	0.865	20
Band 9 – Water vapour	0.945	60
Band 10 – SWIR – Cirrus	1.375	60
Band 11 – SWIR	1.610	20
Band 12 – SWIR	2.190	20

(B4, B3, B2 – 10 m)

(B8a, B11, B12 - 20 m)

Multispectral multiresolution imaging (Sentinel 2)

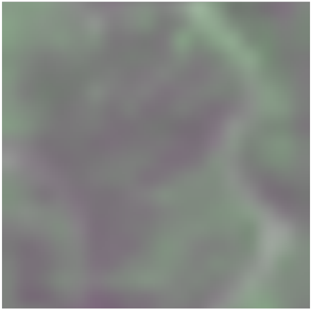
Sentinel-2 Bands	Central Wavelength (µm)	Resolution (m)
Band 1 – Coastal aerosol	0.443	60
Band 2 – Blue	0.490	10
Band 3 – Green	0.560	10
Band 4 – Red	0.665	10
Band 5 – Vegetation Red Edge	0.705	20
Band 6 – Vegetation Red Edge	0.740	20

Sentinel-2 Bands	Central Wavelength (µm)	Resolution (m)
Band 7 – Vegetation Red Edge	0.783	20
Band 8 – NIR	0.842	10
Band 8A – Narrow NIR	0.865	20
Band 9 – Water vapour	0.945	60
Band 10 – SWIR – Cirrus	1.375	60
Band 11 – SWIR	1.610	20
Band 12 – SWIR	2.190	20

(B4, B3, B2 – 10 m)

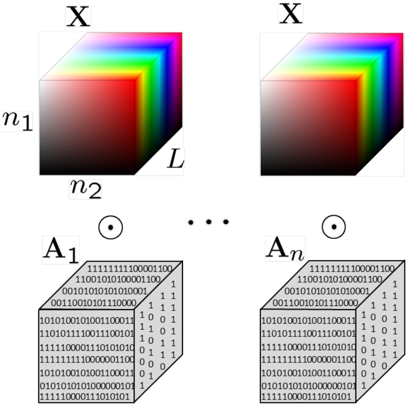
(B8a, B11, B12 - 20 m)

(B1,B9 – 60 m)



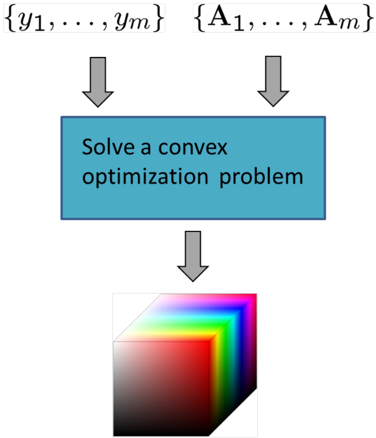
Hyperspectral compressive sensing

Hyperspectral compressive sensing



$y_1 = \langle \mathbf{X}, \mathbf{A}_1 \rangle$ $\quad$ $y_m = \langle \mathbf{X}, \mathbf{A}_1 \rangle$

$m \ll n_1 \times n_2 \times L$



perfect reconstruction

Curing Ill-posed/Ill-conditioned inverse problems

Curing Ill-posed/Ill-conditioned inverse problems

Golden rule for solving ill-posed/ill-conditioned inverse problems

Search for solutions which are:

Curing Ill-posed/Ill-conditioned inverse problems

Golden rule for solving ill-posed/ill-conditioned inverse problems

Search for solutions which are:

1. compatible with the observed data

Curing Ill-posed/Ill-conditioned inverse problems

Golden rule for solving ill-posed/ill-conditioned inverse problems

Search for solutions which are:

1. compatible with the observed data
2. satisfy additional constraints (*a priori* or prior information) coming from the (physics) problem

Curing Ill-posed/Ill-conditioned inverse problems

Golden rule for solving ill-posed/ill-conditioned inverse problems

Search for solutions which are:

1. compatible with the observed data

2. satisfy additional constraints (*a priori* or prior information) coming from the (physics) problem

Frameworks to solve inverse problems

Curing Ill-posed/Ill-conditioned inverse problems

Golden rule for solving ill-posed/ill-conditioned inverse problems

Search for solutions which are:

1. compatible with the observed data
2. satisfy additional constraints (*a priori* or prior information) coming from the (physics) problem

Frameworks to solve inverse problems

- Bayesian inference: the causes are inferred by minimizing the Bayesian risk

Curing Ill-posed/Ill-conditioned inverse problems

Golden rule for solving ill-posed/ill-conditioned inverse problems

Search for solutions which are:

1. compatible with the observed data
2. satisfy additional constraints (*a priori* or prior information) coming from the (physics) problem

Frameworks to solve inverse problems

- Bayesian inference: the causes are inferred by minimizing the Bayesian risk
- Variational regularization: the causes are inferred by minimizing a cost function

Variational regularization framework

Variational regularization framework

Unconstrained formulation: $\min_{\mathbf{x}} f(\mathbf{x}, \mathbf{y}) + \tau \phi(\mathbf{x})$

Variational regularization framework

Unconstrained formulation: $\min_{\mathbf{x}} f(\mathbf{x}, \mathbf{y}) + \tau\phi(\mathbf{x})$

- $f(\mathbf{x}, \mathbf{y}) \rightarrow$ data fidelity term: measures the compatibility between $\mathbf{x}$ and $\mathbf{y}$ (data-term, loss function, observation model, log-likelihood, ...)

Variational regularization framework

Unconstrained formulation: $\min_{\mathbf{x}} f(\mathbf{x}, \mathbf{y}) + \tau\phi(\mathbf{x})$

- $f(\mathbf{x}, \mathbf{y}) \rightarrow$ data fidelity term: measures the compatibility between $\mathbf{x}$ and $\mathbf{y}$ (data-term, loss function, observation model, log-likelihood, ...)
- $\phi(\mathbf{x}) \rightarrow$ regularizer: expresses prior information about $\mathbf{x}$

Variational regularization framework

Unconstrained formulation: $\min_{\mathbf{x}} f(\mathbf{x}, \mathbf{y}) + \tau\phi(\mathbf{x})$

- $f(\mathbf{x}, \mathbf{y}) \rightarrow$ data fidelity term: measures the compatibility between $\mathbf{x}$ and $\mathbf{y}$ (data-term, loss function, observation model, log-likelihood, ...)
- $\phi(\mathbf{x}) \rightarrow$ regularizer: expresses prior information about $\mathbf{x}$
- $\tau \rightarrow$ regularization parameter: sets the relative weight between the data term and the regularizer

Variational regularization framework

Unconstrained formulation: $\min_{\mathbf{x}} f(\mathbf{x}, \mathbf{y}) + \tau\phi(\mathbf{x})$

- $f(\mathbf{x}, \mathbf{y}) \rightarrow$ data fidelity term: measures the compatibility between $\mathbf{x}$ and $\mathbf{y}$ (data-term, loss function, observation model, log-likelihood, ...)
- $\phi(\mathbf{x}) \rightarrow$ regularizer: expresses prior information about $\mathbf{x}$
- $\tau \rightarrow$ regularization parameter: sets the relative weight between the data term and the regularizer

Proximal algorithms for solving convex inverse problems
SALSA ([Afonso, B-D, Figueiredo, 09, 10]) FBPD ([Condat, 13], [Vũ,13])

Variational regularization framework

Unconstrained formulation: $\min_{\mathbf{x}} f(\mathbf{x}, \mathbf{y}) + \tau\phi(\mathbf{x})$

- $f(\mathbf{x}, \mathbf{y}) \rightarrow$ data fidelity term: measures the compatibility between $\mathbf{x}$ and $\mathbf{y}$ (data-term, loss function, observation model, log-likelihood, ...)
- $\phi(\mathbf{x}) \rightarrow$ regularizer: expresses prior information about $\mathbf{x}$
- $\tau \rightarrow$ regularization parameter: sets the relative weight between the data term and the regularizer

Proximal algorithms for solving convex inverse problems
SALSA ([Afonso, B-D, Figueiredo, 09, 10]) FBPD ([Condat, 13], [Vũ,13])

- new class of iterative methods suited to solve large scale non-smooth convex optimization problems

Variational regularization framework

Unconstrained formulation: $\min_{\mathbf{x}} f(\mathbf{x}, \mathbf{y}) + \tau\phi(\mathbf{x})$

- $f(\mathbf{x}, \mathbf{y}) \rightarrow$ data fidelity term: measures the compatibility between $\mathbf{x}$ and $\mathbf{y}$ (data-term, loss function, observation model, log-likelihood, ...)
- $\phi(\mathbf{x}) \rightarrow$ regularizer: expresses prior information about $\mathbf{x}$
- $\tau \rightarrow$ regularization parameter: sets the relative weight between the data term and the regularizer

Proximal algorithms for solving convex inverse problems

SALSA ([Afonso, B-D, Figueiredo, 09, 10]) FBPD ([Condat, 13], [Vũ,13])

- new class of iterative methods suited to solve large scale non-smooth convex optimization problems
- replace a difficult problem with a sequence of simpler ones

Variational regularization framework

Unconstrained formulation: $\min_{\mathbf{x}} f(\mathbf{x}, \mathbf{y}) + \tau \phi(\mathbf{x})$

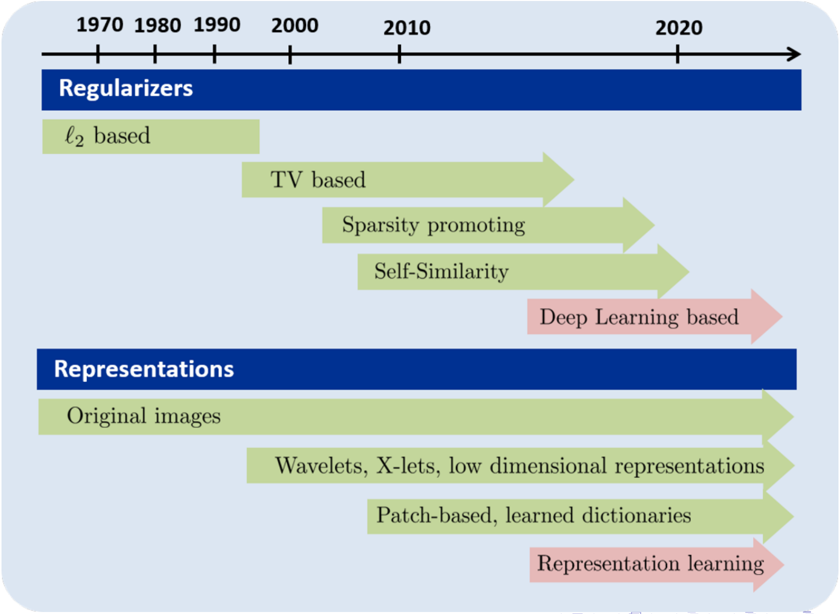
- $f(\mathbf{x}, \mathbf{y}) \rightarrow$ data fidelity term: measures the compatibility between $\mathbf{x}$ and $\mathbf{y}$ (data-term, loss function, observation model, log-likelihood, ...)
- $\phi(\mathbf{x}) \rightarrow$ regularizer: expresses prior information about $\mathbf{x}$
- $\tau \rightarrow$ regularization parameter: sets the relative weight between the data term and the regularizer

Proximal algorithms for solving convex inverse problems
SALSA ([Afonso, B-D, Figueiredo, 09, 10]) FBPD ([Condat, 13], [Vũ,13])

- new class of iterative methods suited to solve large scale non-smooth convex optimization problems
- replace a difficult problem with a sequence of simpler ones
- proximity operators, which may be interpreted as implicit subgradients, plays a central role in the proximal algorithms

Image regularizers and representations

Image regularizers and representations



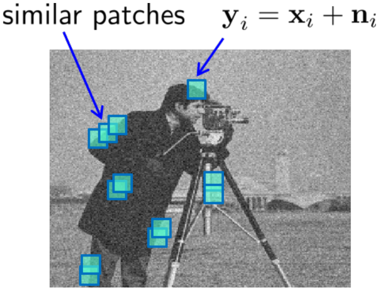
Non-local patch(cube)-based methods

Non-local patch(cube)-based methods

- Real world images are self-similar: given an image patch (cube in volumes), there are similar patches at different locations and scales

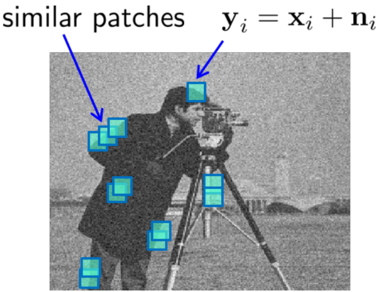
Non-local patch(cube)-based methods

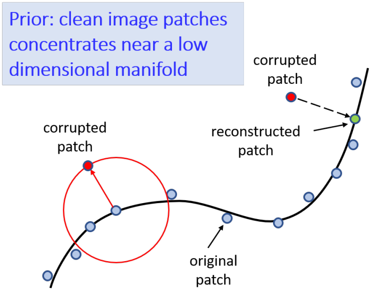
- Real world images are self-similar: given an image patch (cube in volumes), there are similar patches at different locations and scales



Non-local patch(cube)-based methods

- Real world images are self-similar: given an image patch (cube in volumes), there are similar patches at different locations and scales





Patch-based denoising

- Rather than globally modeling $\mathbf{x}$, model patches: divide and conquer

Patch-based denoising

- Rather than globally modeling $\mathbf{x}$, model patches: divide and conquer
  - ✓ Decompose noisy image into (overlapping) patches $\mathbf{y}_i$

Patch-based denoising

- Rather than globally modeling $\mathbf{x}$, model patches: divide and conquer
  - ✓ Decompose noisy image into (overlapping) patches $\mathbf{y}_i$
  - ✓ Denoise each patch: $\hat{\mathbf{x}}_i = \text{denoiser}(\mathbf{y}_i)$

Patch-based denoising

- Rather than globally modeling $\mathbf{x}$, model patches: divide and conquer
  - ✓ Decompose noisy image into (overlapping) patches $\mathbf{y}_i$
  - ✓ Denoise each patch: $\hat{\mathbf{x}}_i = \text{denoiser}(\mathbf{y}_i)$
  - ✓ Average overlapping pixel estimates

Patch-based denoising

- Rather than globally modeling $\mathbf{x}$, model patches: divide and conquer
  - ✓ Decompose noisy image into (overlapping) patches $\mathbf{y}_i$
  - ✓ Denoise each patch: $\hat{\mathbf{x}}_i = \text{denoiser}(\mathbf{y}_i)$
  - ✓ Average overlapping pixel estimates
- Self-similarity has been exploited in various ways

Patch-based denoising

- Rather than globally modeling $\mathbf{x}$, model patches: divide and conquer
  - ✓ Decompose noisy image into (overlapping) patches $\mathbf{y}_i$
  - ✓ Denoise each patch: $\hat{\mathbf{x}}_i = \text{denoiser}(\mathbf{y}_i)$
  - ✓ Average overlapping pixel estimates
- Self-similarity has been exploited in various ways
  - ✓ Non-local (generalized) means: $\widehat{\mathbf{x}}_i = f(\mathbf{y}_{i_1}, \ldots, \mathbf{y}_{i_P})$ ($\mathbf{y}_{i_k}$– similar patches) [Buades et al., 05; Dabov et al., 07; Chatterjee & Milanfar, 09; Maggioni et al., 12; Lebrun et al., 13, Rajwade et al., 13, ...]

Patch-based denoising

- Rather than globally modeling $\mathbf{x}$, model patches: divide and conquer
  - ✓ Decompose noisy image into (overlapping) patches $\mathbf{y}_i$
  - ✓ Denoise each patch: $\hat{\mathbf{x}}_i = \text{denoiser}(\mathbf{y}_i)$
  - ✓ Average overlapping pixel estimates
- Self-similarity has been exploited in various ways
  - ✓ Non-local (generalized) means: $\widehat{\mathbf{x}}_i = f(\mathbf{y}_{i_1}, \ldots, \mathbf{y}_{i_P})$ ($\mathbf{y}_{i_k}$– similar patches) [Buades et al., 05; Dabov et al., 07; Chatterjee & Milanfar, 09; Maggioni et al., 12; Lebrun et al., 13, Rajwade et al., 13, ...]
  - ✓ Dictionary learning: $$\min_{\mathbf{D}, \boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{N_p}} \sum_{i=1}^{N_p} \|\mathbf{y}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda \|\boldsymbol{\alpha}_i\|_1$$ [Elad & Aharon, 05; Mairal et. al., 08,10],

Patch-based denoising

- Rather than globally modeling $\mathbf{x}$, model patches: divide and conquer

  ✓ Decompose noisy image into (overlapping) patches $\mathbf{y}_i$

  ✓ Denoise each patch: $\hat{\mathbf{x}}_i = \text{denoiser}(\mathbf{y}_i)$

  ✓ Average overlapping pixel estimates

- Self-similarity has been exploited in various ways

  ✓ Non-local (generalized) means: $\widehat{\mathbf{x}}_i = f(\mathbf{y}_{i_1}, \ldots, \mathbf{y}_{i_P})$ ($\mathbf{y}_{i_k}$– similar patches)
  [Buades et al., 05; Dabov et al., 07; Chatterjee & Milanfar, 09; Maggioni et al., 12; Lebrun et al., 13, Rajwade et al., 13, ...]

  ✓ Dictionary learning: $$\min_{\mathbf{D},\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_{N_p}} \sum_{i=1}^{N_p} \|\mathbf{y}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda\|\boldsymbol{\alpha}_i\|_1$$
  [Elad & Aharon, 05; Mairal et. al., 08,10],

  - Gaussian mixture models (GMM) and MMSE estimates:

$$\widehat{\mathbf{x}}_i = \mathbb{E}[\mathbf{x}|\mathbf{y}_i] = \int \mathbf{x}\frac{p(\mathbf{y}_i|\mathbf{x})p(\mathbf{x})}{p(\mathbf{y}_i)}\,d\mathbf{x}$$

  [Zoran and Y. Weiss, 11;Yu et al., 12; Teodoro et al., 16; Houdard et al., 17]

Patch MMSE denoising

- Questions:

Patch MMSE denoising

- Questions:
  - ✓ How to choose/learn a prior?

Patch MMSE denoising

- Questions:
  - ✓ How to choose/learn a prior?
  - ✓ How to compute the expectation?

Patch MMSE denoising

- Questions:
  - ✓ How to choose/learn a prior?
  - ✓ How to compute the expectation?
- Essentially three classes of approaches:

Patch MMSE denoising

- Questions:
  - ✓ How to choose/learn a prior?
  - ✓ How to compute the expectation?
- Essentially three classes of approaches:
  - ✓ Non-parametric:

Patch MMSE denoising

- Questions:
  - ✓ How to choose/learn a prior?
  - ✓ How to compute the expectation?
- Essentially three classes of approaches:
  - ✓ Non-parametric:
    - non-local means [Buades et al., 05] and its many variants/descendants;

Patch MMSE denoising

- Questions:
  - ✓ How to choose/learn a prior?
  - ✓ How to compute the expectation?
- Essentially three classes of approaches:
  - ✓ Non-parametric:
    - non-local means [Buades et al., 05] and its many variants/descendants;
    - importance sampling [Niknejad et al., 17]

Patch MMSE denoising

- Questions:
  - ✓ How to choose/learn a prior?
  - ✓ How to compute the expectation?
- Essentially three classes of approaches:
  - ✓ Non-parametric:
    - non-local means [Buades et al., 05] and its many variants/descendants;
    - importance sampling [Niknejad et al., 17]
  - ✓ Semi-parametric: Gaussian mixture models [Zoran and Weiss, 2011, Teodoro et al., 2015, Houdard et al., 2017]

Patch MMSE denoising

- Questions:
  - ✓ How to choose/learn a prior?
  - ✓ How to compute the expectation?
- Essentially three classes of approaches:
  - ✓ Non-parametric:
    - non-local means [Buades et al., 05] and its many variants/descendants;
    - importance sampling [Niknejad et al., 17]
  - ✓ Semi-parametric: Gaussian mixture models [Zoran and Weiss, 2011, Teodoro et al., 2015, Houdard et al., 2017]
  - ✓ Parametric: non-local Bayesian [Lebrun et al., 13] and variants thereof [Niknejad et al., 15, Aguerrebere et al., 17]

Patch MMSE Denoising: Gaussian Prior

- Classical result in linear estimation (Wiener filter)

$$\left.\begin{aligned} p(\mathbf{y}|\mathbf{x}) &= \mathcal{N}(\mathbf{y};\mathbf{x},\sigma^2\mathbf{I}) \\ p(\mathbf{x}) &= \mathcal{N}(\mathbf{x};\boldsymbol{\mu},\mathbf{C}) \end{aligned}\right\}$$

Patch MMSE Denoising: Gaussian Prior

- Classical result in linear estimation (Wiener filter)

$$\left.\begin{aligned} p(\mathbf{y}|\mathbf{x}) &= \mathcal{N}(\mathbf{y}; \mathbf{x}, \sigma^2\mathbf{I}) \\ p(\mathbf{x}) &= \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \mathbf{C}) \end{aligned}\right\} \Rightarrow \mathbb{E}[\mathbf{X}|\mathbf{y}] = \boldsymbol{\mu} + \mathbf{C}\big(\mathbf{C} + \sigma^2\mathbf{I}\big)^{-1}\big(\mathbf{y} - \boldsymbol{\mu}\big)$$

Patch MMSE Denoising: Gaussian Prior

- Classical result in linear estimation (Wiener filter)

$$\left.\begin{array}{lcl} p(\mathbf{y}|\mathbf{x}) & = & \mathcal{N}(\mathbf{y};\mathbf{x},\sigma^2\mathbf{I}) \\ p(\mathbf{x}) & = & \mathcal{N}(\mathbf{x};\boldsymbol{\mu},\mathbf{C}) \end{array}\right\} \Rightarrow \mathbb{E}[\mathbf{X}|\mathbf{y}] = \boldsymbol{\mu} + \mathbf{C}\big(\mathbf{C}+\sigma^2\mathbf{I}\big)^{-1}(\mathbf{y}-\boldsymbol{\mu})$$

- Basic idea: estimate $\boldsymbol{\mu}$ and $\mathbf{C}$ from (clean or noisy) patches;

Patch MMSE Denoising: Gaussian Prior

- Classical result in linear estimation (Wiener filter)

$$\left.\begin{array}{lcl} p(\mathbf{y}|\mathbf{x}) & = & \mathcal{N}(\mathbf{y};\mathbf{x},\sigma^2\mathbf{I}) \\ p(\mathbf{x}) & = & \mathcal{N}(\mathbf{x};\boldsymbol{\mu},\mathbf{C}) \end{array}\right\} \Rightarrow \mathbb{E}[\mathbf{X}|\mathbf{y}] = \boldsymbol{\mu} + \mathbf{C}\big(\mathbf{C}+\sigma^2\mathbf{I}\big)^{-1}\big(\mathbf{y}-\boldsymbol{\mu}\big)$$

- Basic idea: estimate $\boldsymbol{\mu}$ and $\mathbf{C}$ from (clean or noisy) patches;
- In general, image patches are **not** well modelled by **one** Gaussian

Patch MMSE Denoising: Gaussian Prior

- Classical result in linear estimation (Wiener filter)

$$\left.\begin{array}{lll} p(\mathbf{y}|\mathbf{x}) & = & \mathcal{N}(\mathbf{y};\mathbf{x},\sigma^2\mathbf{I}) \\ p(\mathbf{x}) & = & \mathcal{N}(\mathbf{x};\boldsymbol{\mu},\mathbf{C}) \end{array}\right\} \Rightarrow \mathbb{E}[\mathbf{X}|\mathbf{y}] = \boldsymbol{\mu} + \mathbf{C}\big(\mathbf{C}+\sigma^2\mathbf{I}\big)^{-1}(\mathbf{y}-\boldsymbol{\mu})$$

- Basic idea: estimate $\boldsymbol{\mu}$ and $\mathbf{C}$ from (clean or noisy) patches;
- In general, image patches are **not** well modelled by **one** Gaussian
- Divide and conquer again!

Patch MMSE Denoising: Gaussian Prior

- Classical result in linear estimation (Wiener filter)

$$\left.\begin{array}{lcl} p(\mathbf{y}|\mathbf{x}) & = & \mathcal{N}(\mathbf{y};\mathbf{x},\sigma^2\mathbf{I}) \\ p(\mathbf{x}) & = & \mathcal{N}(\mathbf{x};\boldsymbol{\mu},\mathbf{C}) \end{array}\right\} \Rightarrow \mathbb{E}[\mathbf{X}|\mathbf{y}] = \boldsymbol{\mu} + \mathbf{C}\big(\mathbf{C}+\sigma^2\mathbf{I}\big)^{-1}(\mathbf{y}-\boldsymbol{\mu})$$

- Basic idea: estimate $\boldsymbol{\mu}$ and $\mathbf{C}$ from (clean or noisy) patches;
- In general, image patches are **not** well modelled by **one** Gaussian
- Divide and conquer again!
  - ✓ Use only patches similar enough to the patch being denoised [Lebrun et al., 13, Niknejad et al., 15, Aguerrebere et al., 17]

Patch MMSE Denoising: Gaussian Prior

- Classical result in linear estimation (Wiener filter)

$$\left.\begin{array}{rcl} p(\mathbf{y}|\mathbf{x}) & = & \mathcal{N}(\mathbf{y};\mathbf{x},\sigma^2\mathbf{I}) \\ p(\mathbf{x}) & = & \mathcal{N}(\mathbf{x};\boldsymbol{\mu},\mathbf{C}) \end{array}\right\} \Rightarrow \mathbb{E}[\mathbf{X}|\mathbf{y}] = \boldsymbol{\mu} + \mathbf{C}\big(\mathbf{C}+\sigma^2\mathbf{I}\big)^{-1}(\mathbf{y}-\boldsymbol{\mu})$$

- Basic idea: estimate $\boldsymbol{\mu}$ and $\mathbf{C}$ from (clean or noisy) patches;
- In general, image patches are **not** well modelled by **one** Gaussian
- Divide and conquer again!
  - ✓ Use only patches similar enough to the patch being denoised [Lebrun et al., 13, Niknejad et al., 15, Aguerrebere et al., 17]
  - ✓ Use a Gaussian mixture model (GMM)
    - learned from an external set of clean patches [Zoran and Weiss, 11]
    - ... or from the noisy patches [Teodoro et al., 15, Houdard et al., 17]

Patch MMSE denoising: GMM prior

- Previous result extends to GMM priors

$$\left.\begin{aligned} p(\mathbf{y}|\mathbf{x}) &= \mathcal{N}(\mathbf{x}, \sigma^2\mathbf{I}) \\ p(\mathbf{x}) &= \sum_{m=1}^{K} \alpha_m \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_m, \mathbf{C}_m) \end{aligned}\right\}$$

Patch MMSE denoising: GMM prior

- Previous result extends to GMM priors

$$\left.\begin{aligned} p(\mathbf{y}|\mathbf{x}) &= \mathcal{N}(\mathbf{x}, \sigma^2\mathbf{I}) \\ p(\mathbf{x}) &= \sum_{m=1}^{K} \alpha_m \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_m, \mathbf{C}_m) \end{aligned}\right\} \Rightarrow \mathbb{E}[\mathbf{X}|\mathbf{y}] = \frac{\sum_{m=1}^{K} \beta_m(\mathbf{y})\, \mathbf{v}_m(\mathbf{y})}{\sum_{m=1}^{K} \beta_m(\mathbf{y})}$$

Patch MMSE denoising: GMM prior

- Previous result extends to GMM priors

$$\left.\begin{array}{lcl} p(\mathbf{y}|\mathbf{x}) & = & \mathcal{N}(\mathbf{x}, \sigma^2 \mathbf{I}) \\ p(\mathbf{x}) & = & \sum_{m=1}^{K} \alpha_m \, \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_m, \mathbf{C}_m) \end{array}\right\} \Rightarrow \mathbb{E}[\mathbf{X}|\mathbf{y}] = \frac{\sum_{m=1}^{K} \beta_m(\mathbf{y}) \, \mathbf{v}_m(\mathbf{y})}{\sum_{m=1}^{K} \beta_m(\mathbf{y})}$$

where $\beta_m(\mathbf{y}) / \sum_m \beta_m(\mathbf{y}) =$ posterior probability of component m, given $\mathbf{y}$

$$\beta_m(\mathbf{y}) = \alpha_m \, \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_m, \mathbf{C}_m + \sigma^2 \mathbf{I})$$

Patch MMSE denoising: GMM prior

- Previous result extends to GMM priors

$$\left.\begin{aligned} p(\mathbf{y}|\mathbf{x}) &= \mathcal{N}(\mathbf{x}, \sigma^2\mathbf{I}) \\ p(\mathbf{x}) &= \sum_{m=1}^{K} \alpha_m \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_m, \mathbf{C}_m) \end{aligned}\right\} \Rightarrow \mathbb{E}[\mathbf{X}|\mathbf{y}] = \frac{\sum_{m=1}^{K} \beta_m(\mathbf{y})\, \mathbf{v}_m(\mathbf{y})}{\sum_{m=1}^{K} \beta_m(\mathbf{y})}$$

where $\beta_m(\mathbf{y})/\sum_m \beta_m(\mathbf{y})$ = posterior probability of component m, given $\mathbf{y}$

$$\beta_m(\mathbf{y}) = \alpha_m \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_m, \mathbf{C}_m + \sigma^2\mathbf{I})$$

and the $\mathbf{v}_m(\mathbf{y})$ are the component-wise MMSE/Wiener estimates

$$\mathbf{v}_m(\mathbf{y}) = \boldsymbol{\mu}_m + \mathbf{C}_m\left(\mathbf{C}_m + \sigma^2\mathbf{I}\right)^{-1}(\mathbf{y} - \boldsymbol{\mu}_m)$$

Patch MMSE denoising: GMM prior

- Previous result extends to GMM priors

$$\left.\begin{array}{rcl} p(\mathbf{y}|\mathbf{x}) & = & \mathcal{N}(\mathbf{x}, \sigma^2\mathbf{I}) \\ p(\mathbf{x}) & = & \displaystyle\sum_{m=1}^{K} \alpha_m \, \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_m, \mathbf{C}_m) \end{array}\right\} \Rightarrow \mathbb{E}[\mathbf{X}|\mathbf{y}] = \frac{\sum_{m=1}^{K} \beta_m(\mathbf{y}) \, \mathbf{v}_m(\mathbf{y})}{\sum_{m=1}^{K} \beta_m(\mathbf{y})}$$

where $\beta_m(\mathbf{y}) / \sum_m \beta_m(\mathbf{y}) =$ posterior probability of component m, given $\mathbf{y}$

$$\beta_m(\mathbf{y}) = \alpha_m \, \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_m, \mathbf{C}_m + \sigma^2\mathbf{I})$$

and the $\mathbf{v}_m(\mathbf{y})$ are the component-wise MMSE/Wiener estimates

$$\mathbf{v}_m(\mathbf{y}) = \boldsymbol{\mu}_m + \mathbf{C}_m \left(\mathbf{C}_m + \sigma^2\mathbf{I}\right)^{-1} (\mathbf{y} - \boldsymbol{\mu}_m)$$

- Interestingly, the MAP estimate is hard to find: it is **not** the mode of the most probable component [Carreira-Perpiñán, 02]

Patch MMSE denoising: GMM prior

- Previous result extends to GMM priors

$$\left.\begin{array}{rcl} p(\mathbf{y}|\mathbf{x}) & = & \mathcal{N}(\mathbf{x}, \sigma^2 \mathbf{I}) \\ p(\mathbf{x}) & = & \displaystyle\sum_{m=1}^{K} \alpha_m \, \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_m, \mathbf{C}_m) \end{array}\right\} \Rightarrow \mathbb{E}[\mathbf{X}|\mathbf{y}] = \frac{\sum_{m=1}^{K} \beta_m(\mathbf{y}) \, \mathbf{v}_m(\mathbf{y})}{\sum_{m=1}^{K} \beta_m(\mathbf{y})}$$

where $\beta_m(\mathbf{y}) / \sum_m \beta_m(\mathbf{y})$ = posterior probability of component m, given $\mathbf{y}$

$$\beta_m(\mathbf{y}) = \alpha_m \, \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_m, \mathbf{C}_m + \sigma^2 \mathbf{I})$$

and the $\mathbf{v}_m(\mathbf{y})$ are the component-wise MMSE/Wiener estimates

$$\mathbf{v}_m(\mathbf{y}) = \boldsymbol{\mu}_m + \mathbf{C}_m \left(\mathbf{C}_m + \sigma^2 \mathbf{I}\right)^{-1} (\mathbf{y} - \boldsymbol{\mu}_m)$$

- Interestingly, the MAP estimate is hard to find: it is **not** the mode of the most probable component [Carreira-Perpiñán, 02]
- Can also get var$[x_i|\mathbf{y}]$: use (inverse) as weight in assembling the image estimate from the patch estimates

Patch MMSE Denoising: GMM Prior

- Learning the GMM:

Patch MMSE Denoising: GMM Prior

- Learning the GMM:
  - ✓ From external set of clean images: trivial via EM [Zoran and Weiss, 2011]

Patch MMSE Denoising: GMM Prior

- Learning the GMM:
  - ✓ From external set of clean images: trivial via EM [Zoran and Weiss, 2011]
  - ✓ Directly from the noisy patches

$$(\hat{\alpha}_m, \hat{\boldsymbol{\mu}}_m, \hat{\mathbf{C}}_m)_{m=1,\ldots,K} = \arg\max \sum_{i=1}^{M} \log \sum_{m=1}^{K} \alpha_m \mathcal{N}(\mathbf{y}_i; \boldsymbol{\mu}_m, \mathbf{C}_m + \sigma^2 \mathbf{I})$$

$$\text{subject to } \mathbf{C}_m \succeq 0$$

Patch MMSE Denoising: GMM Prior

- Learning the GMM:
  - ✓ From external set of clean images: trivial via EM [Zoran and Weiss, 2011]
  - ✓ Directly from the noisy patches

$$(\hat{\alpha}_m, \hat{\boldsymbol{\mu}}_m, \hat{\mathbf{C}}_m)_{m=1,\ldots,K} = \arg\max \sum_{i=1}^{M} \log \sum_{m=1}^{K} \alpha_m \mathcal{N}(\mathbf{y}_i; \boldsymbol{\mu}_m, \mathbf{C}_m + \sigma^2 \mathbf{I})$$

$$\text{subject to } \mathbf{C}_m \succeq 0$$

  - ✓ Standard EM for GMM, with a simple change in the covariance updates (eigenvalue thresholding) [Teodoro et al., 15]

Patch MMSE Denoising: GMM Prior

- Learning the GMM:
  - ✓ From external set of clean images: trivial via EM [Zoran and Weiss, 2011]
  - ✓ Directly from the noisy patches

$$(\hat{\alpha}_m, \hat{\boldsymbol{\mu}}_m, \hat{\mathbf{C}}_m)_{m=1,\dots,K} = \arg\max \sum_{i=1}^{M} \log \sum_{m=1}^{K} \alpha_m \mathcal{N}(\mathbf{y}_i; \boldsymbol{\mu}_m, \mathbf{C}_m + \sigma^2 \mathbf{I})$$

$$\text{subject to } \mathbf{C}_m \succeq 0$$

  - ✓ Standard EM for GMM, with a simple change in the covariance updates (eigenvalue thresholding) [Teodoro et al., 15]
  - ✓ May include other constraints: *e.g.*, Toeplitz for shift-invariance

Patch MMSE Denoising: GMM Prior

- Learning the GMM:
  - ✓ From external set of clean images: trivial via EM [Zoran and Weiss, 2011]
  - ✓ Directly from the noisy patches

$$(\hat{\alpha}_m, \hat{\boldsymbol{\mu}}_m, \hat{\mathbf{C}}_m)_{m=1,\dots,K} = \arg\max \sum_{i=1}^{M} \log \sum_{m=1}^{K} \alpha_m \mathcal{N}(\mathbf{y}_i; \boldsymbol{\mu}_m, \mathbf{C}_m + \sigma^2 \mathbf{I})$$

$$\text{subject to } \mathbf{C}_m \succeq 0$$

  - ✓ Standard EM for GMM, with a simple change in the covariance updates (eigenvalue thresholding) [Teodoro et al., 15]
  - ✓ May include other constraints: *e.g.*, Toeplitz for shift-invariance

- Automatically selects the rank of each covariance: dimension of each subspace (a generalization of MPPCA [Tipping and Bishop, 99])

Patch MMSE Denoising: GMM Prior

- Learning the GMM:
  - ✓ From external set of clean images: trivial via EM [Zoran and Weiss, 2011]
  - ✓ Directly from the noisy patches

$$(\hat{\alpha}_m, \hat{\boldsymbol{\mu}}_m, \hat{\mathbf{C}}_m)_{m=1,\ldots,K} = \arg\max \sum_{i=1}^{M} \log \sum_{m=1}^{K} \alpha_m \mathcal{N}(\mathbf{y}_i; \boldsymbol{\mu}_m, \mathbf{C}_m + \sigma^2 \mathbf{I})$$

$$\text{subject to } \mathbf{C}_m \succeq 0$$

  - ✓ Standard EM for GMM, with a simple change in the covariance updates (eigenvalue thresholding) [Teodoro et al., 15]
  - ✓ May include other constraints: *e.g.*, Toeplitz for shift-invariance

- Automatically selects the rank of each covariance: dimension of each subspace (a generalization of MPPCA [Tipping and Bishop, 99])
- Choosing K: model selection for mixtures [Figueiredo. and Jain, 02]

GMM-based denoising

- Further tricks are need to reach state-of-the-art: “Secrets of image denoising cuisine"[Lebrun et al., 12]

GMM-based denoising

- Further tricks are need to reach state-of-the-art: “Secrets of image denoising cuisine"[Lebrun et al., 12]
- *E.g.*, treat flat patches separately, treat DC separately, repeat, ...

GMM-based denoising

- Further tricks are need to reach state-of-the-art: "Secrets of image denoising cuisine"[Lebrun et al., 12]
- *E.g.*, treat flat patches separately, treat DC separately, repeat, ...
- Denoising experiments [Teodoro et al., 15]

original | noisy (σ=30) | BM3D (28.64dB) | GMM (28.66dB)

σ	Lena (512 × 512)				Cameraman (256 × 256)			
	BM3D	K-SVD	Basic	Improved	BM3D	K-SVD	Basic	Improved
5	38.72	38.53	**38.86**	**38.86**	38.29	37.97	38.57	**38.58**
10	**35.93**	35.55	35.88	35.88	34.18	33.76	34.44	**34.49**
15	**34.27**	33.74	34.11	34.11	31.91	31.54	32.15	**32.21**
20	**33.05**	32.40	32.83	32.84	30.48	30.07	30.64	**30.70**
25	**32.08**	31.34	31.81	31.82	29.45	28.94	29.50	**29.58**
30	**31.26**	30.46	30.99	31.00	28.64	28.12	28.58	**28.66**

Hyperspectral denoising

Hyperspectral denoising

- HSIs are low-rank: $\mathbf{Z} = \mathbf{E}\mathbf{X} \in \mathbb{R}^{L \times N}, \quad \mathbf{E} \in \mathbb{R}^{L \times p}, \quad \mathbf{X} \in \mathbb{R}^{p \times N}, \quad p << L$

Hyperspectral denoising

- HSIs are low-rank: $\mathbf{Z} = \mathbf{E}\mathbf{X} \in \mathbb{R}^{L \times N}, \quad \mathbf{E} \in \mathbb{R}^{L \times p}, \quad \mathbf{X} \in \mathbb{R}^{p \times N}, \quad p << L$
- $\mathbf{E}$ may be estimated from the noisy data [B-D,Nascimento, 08])

Hyperspectral denoising

- HSIs are low-rank: $\mathbf{Z} = \mathbf{E}\mathbf{X} \in \mathbb{R}^{L\times N}, \quad \mathbf{E} \in \mathbb{R}^{L\times p}, \quad \mathbf{X} \in \mathbb{R}^{p\times N}, \quad p << L$
- $\mathbf{E}$ may be estimated from the noisy data [B-D,Nascimento, 08])
- HSIs are self-similar $\Rightarrow$ $\mathbf{X}$ (eigen-images) are self-similar

Hyperspectral denoising

- HSIs are low-rank: $\mathbf{Z} = \mathbf{E}\mathbf{X} \in \mathbb{R}^{L\times N}, \quad \mathbf{E} \in \mathbb{R}^{L\times p}, \quad \mathbf{X} \in \mathbb{R}^{p\times N}, \quad p << L$
- $\mathbf{E}$ may be estimated from the noisy data [B-D,Nascimento, 08])
- HSIs are self-similar $\Rightarrow$ $\mathbf{X}$ (eigen-images) are self-similar
- FastHyDe HSI denoiser] [Zhuang, B-D, 16]

$$\begin{aligned}\widehat{\mathbf{Z}} &= \arg\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{E}\mathbf{X} - \mathbf{Y}\|_F^2 + \lambda\phi(\mathbf{X}), \qquad \mathbf{E}^T\mathbf{E} = \mathbf{I} \\ &= \arg\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{X} - \mathbf{E}^T\mathbf{Y}\|_F^2 + \lambda\phi(\mathbf{X})\end{aligned}$$

Hyperspectral denoising

- HSIs are low-rank: $\mathbf{Z} = \mathbf{E}\mathbf{X} \in \mathbb{R}^{L\times N}, \quad \mathbf{E} \in \mathbb{R}^{L\times p}, \quad \mathbf{X} \in \mathbb{R}^{p\times N}, \quad p << L$
- $\mathbf{E}$ may be estimated from the noisy data [B-D,Nascimento, 08])
- HSIs are self-similar $\Rightarrow$ $\mathbf{X}$ (eigen-images) are self-similar
- FastHyDe HSI denoiser] [Zhuang, B-D, 16]

$$\begin{aligned}\widehat{\mathbf{Z}} &= \arg\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{E}\mathbf{X} - \mathbf{Y}\|_F^2 + \lambda\phi(\mathbf{X}), \qquad \mathbf{E}^T\mathbf{E} = \mathbf{I} \\ &= \arg\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{X} - \mathbf{E}^T\mathbf{Y}\|_F^2 + \lambda\phi(\mathbf{X})\end{aligned}$$

- Regularizer ϕ is decoupled: $\phi(\mathbf{X}) = \sum_{i=1}^{p} \phi_i(\mathbf{x}^i)$

Hyperspectral denoising

- HSIs are low-rank: $\mathbf{Z} = \mathbf{E}\mathbf{X} \in \mathbb{R}^{L\times N}, \quad \mathbf{E} \in \mathbb{R}^{L\times p}, \quad \mathbf{X} \in \mathbb{R}^{p\times N}, \quad p << L$
- $\mathbf{E}$ may be estimated from the noisy data [B-D,Nascimento, 08])
- HSIs are self-similar $\Rightarrow$ $\mathbf{X}$ (eigen-images) are self-similar
- FastHyDe HSI denoiser] [Zhuang, B-D, 16]

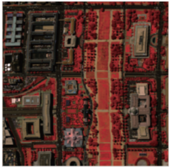
$$\begin{aligned}\widehat{\mathbf{Z}} &= \arg\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{E}\mathbf{X} - \mathbf{Y}\|_F^2 + \lambda\phi(\mathbf{X}), \qquad \mathbf{E}^T\mathbf{E} = \mathbf{I} \\ &= \arg\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{X} - \mathbf{E}^T\mathbf{Y}\|_F^2 + \lambda\phi(\mathbf{X})\end{aligned}$$

- Regularizer ϕ is decoupled: $\phi(\mathbf{X}) = \sum_{i=1}^{p} \phi_i(\mathbf{x}^i)$

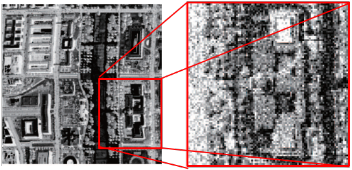
$$\widehat{\mathbf{X}} = \text{prox}_{\lambda\phi}(\mathbf{E}^T\mathbf{Y}) = \begin{bmatrix} \text{prox}_{\lambda\phi_1}(\mathbf{e}_1^T\mathbf{Y}) \\ \vdots \\ \text{prox}_{\lambda\phi_k}(\mathbf{e}_k^T\mathbf{Y}) \end{bmatrix}$$

Hyperspectral denoising ($\text{prox}_{\lambda\phi} \equiv \text{BM3D}$)

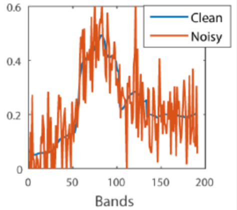
Hyperspectral denoising ($\text{prox}_{\lambda\phi} \equiv \text{BM3D}$)



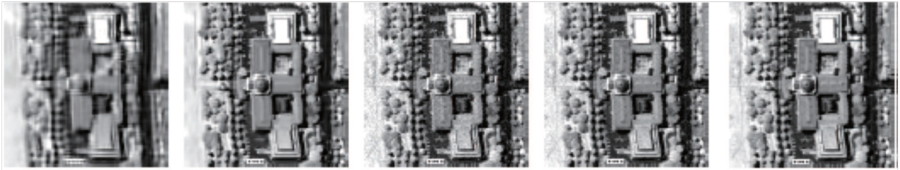
original HSI
$256 \times 256 \times 191$



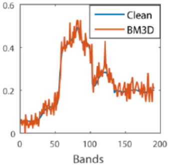
band 70 | noisy band 70



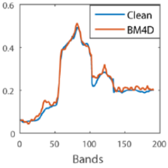
Case 1 (Gaussian i.i.d. noise)



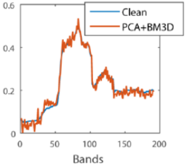
Case 1 BM3D | Case 1 BM4D | Case 1 PCA+BM4D | Case 1 NAILRMA | Case 1 FastHyDe



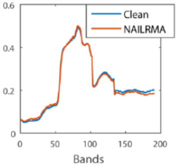
Case 1 (Gaussian i.i.d. noise)



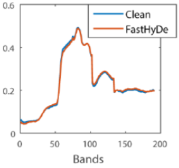
Case 1 (Gaussian i.i.d. noise)



Case 1 (Gaussian i.i.d. noise)



Case 1 (Gaussian i.i.d. noise)



Case 1 (Gaussian i.i.d. noise)

Hyperspectral denoising + inpainting

Hyperspectral denoising + inpainting

$$\widehat{\mathbf{X}} = \arg\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{M}\,\mathsf{vec}(\mathbf{E}\mathbf{X}) - \mathbf{y}\|_F^2 + \lambda\phi(\mathbf{X})$$

Hyperspectral denoising + inpainting

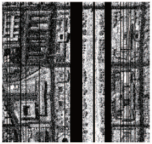
$$\widehat{\mathbf{X}} = \arg\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{M}\,\mathsf{vec}(\mathbf{E}\mathbf{X}) - \mathbf{y}\|_F^2 + \lambda\phi(\mathbf{X})$$

✓ $\mathbf{M}$ is selection matrix

Hyperspectral denoising + inpainting

$$\widehat{\mathbf{X}} = \arg\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{M}\,\mathrm{vec}(\mathbf{E}\mathbf{X}) - \mathbf{y}\|_F^2 + \lambda\phi(\mathbf{X})$$

- ✓ $\mathbf{M}$ is selection matrix
- ✓ $\mathbf{y} = \mathrm{vec}(\mathbf{Y})$



Case 1
Noisy image with stripes

Case 1
PDE

Case 1
UBD

Case 1
LRTV

Case 1
FastHyIn

Quantitative assessment of different inpainting algorithms applied to Washington DC Mall.

	Index	Noisy Image	PDE	UBD	LRTV	FastDyIn
Case 1	MPSNR	19.90	20.01	34.52	35.53	**38.58**
	MSSIM	0.4794	0.4831	0.9575	0.9535	**0.9802**
	Time	-	26	23	210	**12**
Case 2	MPSNR	28.43	28.63	35.46	41.88	**51.63**
	MSSIM	0.7414	0.7466	0.9648	0.9853	**0.9987**
	Time	-	35	23	210	**13**
Case 3	MPSNR	26.80	26.99	37.18	40.78	**43.21**
	MSSIM	0.7952	0.7996	0.9780	0.9831	**0.9919**
	Time	-	26	26	179	**24**

Patch-based: general inverse problems

- General inverse problems (non-additive, non-Gaussian, non-diagonal)

$$f(\mathbf{x}, \mathbf{y}) \neq \alpha \|\mathbf{y} - \mathbf{x}\|^2$$

- Questions:

Patch-based: general inverse problems

- General inverse problems (non-additive, non-Gaussian, non-diagonal)

$$f(\mathbf{x}, \mathbf{y}) \neq \alpha \|\mathbf{y} - \mathbf{x}\|^2$$

- Questions:
  - ✓ How to choose/learn a patch-based prior/regularizer?

Patch-based: general inverse problems

- General inverse problems (non-additive, non-Gaussian, non-diagonal)

$$f(\mathbf{x},\mathbf{y}) \neq \alpha \|\mathbf{y}-\mathbf{x}\|^2$$

- Questions:
  - ✓ How to choose/learn a patch-based prior/regularizer?
  - ✓ How to compute the estimates?

Patch-based: general inverse problems

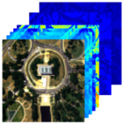
- General inverse problems (non-additive, non-Gaussian, non-diagonal)

$$f(\mathbf{x}, \mathbf{y}) \neq \alpha \|\mathbf{y} - \mathbf{x}\|^2$$

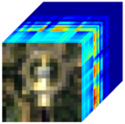
- Questions:
  - ✓ How to choose/learn a patch-based prior/regularizer?
  - ✓ How to compute the estimates?
- Research directions:
  - ✓ Class-adapted patch-based prior/regularizer
  - ✓ Scene-adapted patch-based prior/regularizer
  - ✓ Plug-and-Play: plug a denoiser into the iterations of an iterative solver

Scene adaptation: hyperspectral fusion

- Spectral-spatial resolution trade-off:



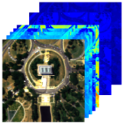
Multi-spectral:
high spatial resolution
low spectral resolution



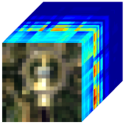
Hyper-spectral:
low spatial resolution
high spectral resolution

Scene adaptation: hyperspectral fusion

- Spectral-spatial resolution trade-off:



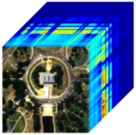
Multi-spectral:
high spatial resolution
low spectral resolution



Hyper-spectral:
low spatial resolution
high spectral resolution

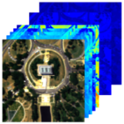
- Fuse MS and HS data:

  high spatial & spectral resolutions

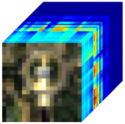


Scene adaptation: hyperspectral fusion

- Spectral-spatial resolution trade-off:



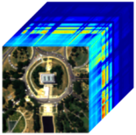
Multi-spectral:
high spatial resolution
low spectral resolution



Hyper-spectral:
low spatial resolution
high spectral resolution

- Fuse MS and HS data:

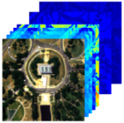
  high spatial & spectral resolutions



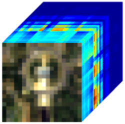
- Extreme case: pansharpening (panchromatic rather than MS image).

Scene adaptation: hyperspectral fusion

- Spectral-spatial resolution trade-off:



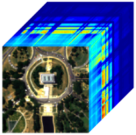
Multi-spectral:
high spatial resolution
low spectral resolution



Hyper-spectral:
low spatial resolution
high spectral resolution

- Fuse MS and HS data:

  high spatial & spectral resolutions



- Extreme case: pansharpening (panchromatic rather than MS image).

Hyperspectral fusion: formulation

- Observation model [Simões et al., 14]

$$\mathbf{Y}_h = \overbrace{\mathbf{E}\mathbf{X}}^{\mathbf{Z}}\mathbf{B}\mathbf{M} + \mathbf{N}_h \qquad \text{hyperspectral data} \in \mathbb{R}^{L_h \times n_h}$$

$$\mathbf{Y}_m = \mathbf{R}\underbrace{\mathbf{E}\mathbf{X}}_{\mathbf{Z}} + \mathbf{N}_m \qquad \text{multispectral data} \in \mathbb{R}^{L_m \times n_m}$$

$L_h > L_m$ and $n_h < n_m$

Hyperspectral fusion: formulation

- Observation model [Simões et al., 14]

$$\mathbf{Y}_h = \overbrace{\mathbf{E}\mathbf{X}}^{\mathbf{Z}}\mathbf{B}\mathbf{M} + \mathbf{N}_h \qquad \text{hyperspectral data} \in \mathbb{R}^{L_h \times n_h}$$

$$\mathbf{Y}_m = \mathbf{R}\underbrace{\mathbf{E}\mathbf{X}}_{\mathbf{Z}} + \mathbf{N}_m \qquad \text{multispectral data} \in \mathbb{R}^{L_m \times n_m}$$

$L_h > L_m$ and $n_h < n_m$

✓ $\mathbf{Z} \in \mathbb{R}^{L_h \times n_m}$: the fused image to be recovered

Hyperspectral fusion: formulation

- Observation model [Simões et al., 14]

$$\mathbf{Y}_h = \overbrace{\mathbf{EX}}^{\mathbf{Z}}\mathbf{BM} + \mathbf{N}_h \qquad \text{hyperspectral data} \in \mathbb{R}^{L_h \times n_h}$$

$$\mathbf{Y}_m = \mathbf{R}\underbrace{\mathbf{EX}}_{\mathbf{Z}} + \mathbf{N}_m \qquad \text{multispectral data} \in \mathbb{R}^{L_m \times n_m}$$

$L_h > L_m$ and $n_h < n_m$

✓ $\mathbf{Z} \in \mathbb{R}^{L_h \times n_m}$: the fused image to be recovered

✓ $\mathbf{E} \in \mathbb{R}^{L_h \times p}$: the p-dimensional subspace containing the fused image $\mathbf{Z}$

Hyperspectral fusion: formulation

- Observation model [Simões et al., 14]

$$\mathbf{Y}_h = \overbrace{\mathbf{E}\mathbf{X}}^{\mathbf{Z}}\mathbf{B}\mathbf{M} + \mathbf{N}_h \qquad \text{hyperspectral data} \in \mathbb{R}^{L_h \times n_h}$$

$$\mathbf{Y}_m = \mathbf{R}\underbrace{\mathbf{E}\mathbf{X}}_{\mathbf{Z}} + \mathbf{N}_m \qquad \text{multispectral data} \in \mathbb{R}^{L_m \times n_m}$$

$L_h > L_m$ and $n_h < n_m$

✓ $\mathbf{Z} \in \mathbb{R}^{L_h \times n_m}$: the fused image to be recovered

✓ $\mathbf{E} \in \mathbb{R}^{L_h \times p}$: the p-dimensional subspace containing the fused image $\mathbf{Z}$

✓ $\mathbf{X} \in \mathbb{R}^{p \times n_h}$: the corresponding coefficients ($p \ll L_h$)

Hyperspectral fusion: formulation

- Observation model [Simões et al., 14]

$$
\begin{aligned}
\mathbf{Y}_h &= \overbrace{\mathbf{E}\mathbf{X}}^{\mathbf{Z}}\,\mathbf{B}\mathbf{M} + \mathbf{N}_h \qquad \text{hyperspectral data} \in \mathbb{R}^{L_h \times n_h} \\
\mathbf{Y}_m &= \mathbf{R}\,\underbrace{\mathbf{E}\mathbf{X}}_{\mathbf{Z}} + \mathbf{N}_m \qquad \text{multispectral data} \in \mathbb{R}^{L_m \times n_m}
\end{aligned}
$$

$L_h > L_m$ and $n_h < n_m$

✓ $\mathbf{Z} \in \mathbb{R}^{L_h \times n_m}$: the fused image to be recovered

✓ $\mathbf{E} \in \mathbb{R}^{L_h \times p}$: the p-dimensional subspace containing the fused image $\mathbf{Z}$

✓ $\mathbf{X} \in \mathbb{R}^{p \times n_h}$: the corresponding coefficients ($p \ll L_h$)

✓ $(\mathbf{B}\,\mathbf{M}) \in \mathbb{R}^{n_m \times n_h}$: spatial convolution & subsampling

Hyperspectral fusion: formulation

- Observation model [Simões et al., 14]

$$\begin{aligned} \mathbf{Y}_h &= \overbrace{\mathbf{E}\mathbf{X}}^{\mathbf{Z}}\mathbf{B}\mathbf{M} + \mathbf{N}_h & \text{hyperspectral data} \in \mathbb{R}^{L_h \times n_h} \\ \mathbf{Y}_m &= \mathbf{R}\underbrace{\mathbf{E}\mathbf{X}}_{\mathbf{Z}} + \mathbf{N}_m & \text{multispectral data} \in \mathbb{R}^{L_m \times n_m} \end{aligned}$$

$L_h > L_m$ and $n_h < n_m$

- ✓ $\mathbf{Z} \in \mathbb{R}^{L_h \times n_m}$: the fused image to be recovered
- ✓ $\mathbf{E} \in \mathbb{R}^{L_h \times p}$: the p-dimensional subspace containing the fused image $\mathbf{Z}$
- ✓ $\mathbf{X} \in \mathbb{R}^{p \times n_h}$: the corresponding coefficients ($p \ll L_h$)
- ✓ $(\mathbf{B}\,\mathbf{M}) \in \mathbb{R}^{n_m \times n_h}$: spatial convolution & subsampling
- ✓ $\mathbf{R} \in \mathbb{R}^{L_m \times L_h}$: spectral responses of the MS sensors

Hyperspectral fusion: formulation

- Observation model [Simões et al., 14]

$$\begin{aligned} \mathbf{Y}_h &= \overbrace{\mathbf{E}\mathbf{X}}^{\mathbf{Z}}\,\mathbf{B}\mathbf{M} + \mathbf{N}_h \\ \mathbf{Y}_m &= \mathbf{R}\,\underbrace{\mathbf{E}\mathbf{X}}_{\mathbf{Z}} + \mathbf{N}_m \end{aligned}$$

hyperspectral data $\in \mathbb{R}^{L_h \times n_h}$

multispectral data $\in \mathbb{R}^{L_m \times n_m}$

$L_h > L_m$ and $n_h < n_m$

✓ $\mathbf{Z} \in \mathbb{R}^{L_h \times n_m}$: the fused image to be recovered

✓ $\mathbf{E} \in \mathbb{R}^{L_h \times p}$: the p-dimensional subspace containing the fused image $\mathbf{Z}$

✓ $\mathbf{X} \in \mathbb{R}^{p \times n_h}$: the corresponding coefficients ($p \ll L_h$)

✓ $(\mathbf{B}\,\mathbf{M}) \in \mathbb{R}^{n_m \times n_h}$: spatial convolution & subsampling

✓ $\mathbf{R} \in \mathbb{R}^{L_m \times L_h}$: spectral responses of the MS sensors

✓ $\mathbf{N}_h$ and $\mathbf{N}_m$: noise

Scene adaptation: dictionary-based regularization

Scene adaptation: dictionary-based regularization

- Hyperspectral-multispectral fusion dictionary-based regularization

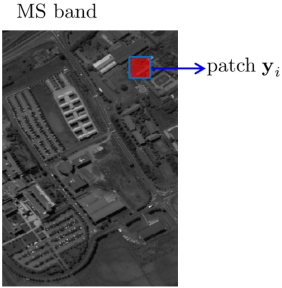
Scene adaptation: dictionary-based regularization

- Hyperspectral-multispectral fusion dictionary-based regularization
- Motivation: patch-based dictionaries learned from the (high spatial resolution) MS bands fit very well the HS bands

Scene adaptation: dictionary-based regularization

- Hyperspectral-multispectral fusion dictionary-based regularization

- Motivation: patch-based dictionaries learned from the (high spatial resolution) MS bands fit very well the HS bands



Scene adaptation: dictionary-based regularization

- Hyperspectral-multispectral fusion dictionary-based regularization

- Motivation: patch-based dictionaries learned from the (high spatial resolution) MS bands fit very well the HS bands

MS band

→ patch $\mathbf{y}_i$

DL →

$$\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_K]$$

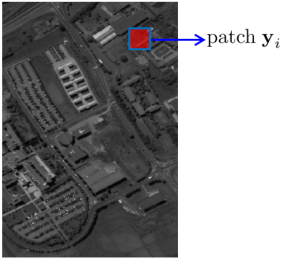
$$\mathbf{y}_i \simeq \sum_{i \in S_i} \alpha_i \mathbf{d}_i$$

Scene adaptation: dictionary-based regularization

- Hyperspectral-multispectral fusion dictionary-based regularization

- Motivation: patch-based dictionaries learned from the (high spatial resolution) MS bands fit very well the HS bands

MS band



DL $\rightarrow$

$$\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_K]$$

$$\mathbf{y}_i \simeq \sum_{i \in S_i} \alpha_i \mathbf{d}_i$$

- A path $\mathbf{z}_i$ of the a HS band is well approximated by the dictionary atoms $\mathbf{d}_i$ for $i \in \mathcal{S}_i$

$$\mathbf{z}_i \simeq \sum_{i \in \mathcal{S}_i} a_i \mathbf{d}_i \qquad \Rightarrow \qquad \mathbf{Z} \simeq \mathcal{L}(\mathbf{D}, \mathbf{A}, \mathcal{S})$$

Hyperspectral fusion

Hyperspectral fusion

- HS-MS image fusion based on a sparse representation (HFSR) [Wei et al., 15]

$$\min_{\mathbf{X},\mathbf{A}} (1/2)\left\|\mathbf{Y}_h - \mathbf{EXBM}\right\|_{Q_h}^2 + (1/2)\left\|\mathbf{Y}_m - \mathbf{REX}\right\|_{Q_m}^2 + \tau\phi_{\mathsf{DL}}(\mathbf{X},\mathbf{A})$$

Hyperspectral fusion

- HS-MS image fusion based on a sparse representation (HFSR) [Wei et al., 15]

$$\min_{\mathbf{X},\mathbf{A}} (1/2)\left\|\mathbf{Y}_h - \mathbf{EXBM}\right\|_{Q_h}^2 + (1/2)\left\|\mathbf{Y}_m - \mathbf{REX}\right\|_{Q_m}^2 + \tau\phi_{\mathsf{DL}}(\mathbf{X},\mathbf{A})$$

✓ $\mathbf{A}$ is the code for $\mathbf{X}$ with respect to the dictionary $\mathbf{D}$

Hyperspectral fusion

- HS-MS image fusion based on a sparse representation (HFSR) [Wei et al., 15]

$$\min_{\mathbf{X},\mathbf{A}} (1/2)\left\|\mathbf{Y}_h - \mathbf{EXBM}\right\|_{Q_h}^2 + (1/2)\left\|\mathbf{Y}_m - \mathbf{REX}\right\|_{Q_m}^2 + \tau\phi_{\mathsf{DL}}(\mathbf{X},\mathbf{A})$$

✓ $\mathbf{A}$ is the code for $\mathbf{X}$ with respect to the dictionary $\mathbf{D}$

$$\phi_{\mathsf{DL}}(\mathbf{X},\mathbf{A}) := \left\|\mathbf{EX} - \mathcal{L}(\mathbf{D},\mathbf{A},\mathcal{S})\right\|_F^2$$

Hyperspectral fusion

- HS-MS image fusion based on a sparse representation (HFSR) [Wei et al., 15]

$$\min_{\mathbf{X},\mathbf{A}} (1/2)\left\|\mathbf{Y}_h - \mathbf{EXBM}\right\|_{Q_h}^2 + (1/2)\left\|\mathbf{Y}_m - \mathbf{REX}\right\|_{Q_m}^2 + \tau\phi_{\mathsf{DL}}(\mathbf{X},\mathbf{A})$$

  ✓ $\mathbf{A}$ is the code for $\mathbf{X}$ with respect to the dictionary $\mathbf{D}$

$$\phi_{\mathsf{DL}}(\mathbf{X},\mathbf{A}) := \left\|\mathbf{EX} - \mathcal{L}(\mathbf{D},\mathbf{A},\mathcal{S})\right\|_F^2$$

  ✓ $\mathcal{S}$ is the support of the code learned from the MS images

Hyperspectral fusion

- HS-MS image fusion based on a sparse representation (HFSR) [Wei et al., 15]

$$\min_{\mathbf{X},\mathbf{A}} (1/2)\left\|\mathbf{Y}_h - \mathbf{EXBM}\right\|_{Q_h}^2 + (1/2)\left\|\mathbf{Y}_m - \mathbf{REX}\right\|_{Q_m}^2 + \tau\phi_{\mathsf{DL}}(\mathbf{X},\mathbf{A})$$

  ✓ $\mathbf{A}$ is the code for $\mathbf{X}$ with respect to the dictionary $\mathbf{D}$

$$\phi_{\mathsf{DL}}(\mathbf{X},\mathbf{A}) := \left\|\mathbf{EX} - \mathcal{L}(\mathbf{D},\mathbf{A},\mathcal{S})\right\|_F^2$$

  ✓ $\mathcal{S}$ is the support of the code learned from the MS images

Algorithm 6: HFSR

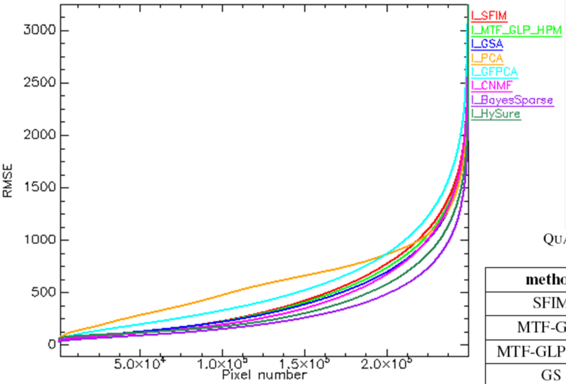
Learn the dictionary using online learning [Mairal et al., 09]
Compute the support $\mathcal{S}$
for $k = 0, 1, \ldots$ **do**
  optimize wrt $\mathbf{X}$ using SALSA [Afonso et al., 11]
  use gradient descent wrt $\mathbf{A}$

Camargue performance indexes

[Loncan et al. , 15]

Camargue performance indexes
[Loncan et al. , 15]

Camargue data set



QUALITY MEASURES FOR THE CAMARGUE DATASET

method	CC	SAM	RMSE	ERGAS
SFIM	0.91886	4.2895	637.1451	3.4159
MTF-GLP	0.92397	4.3378	622.4711	3.2666
MTF-GLP-HPM	0.92599	4.2821	611.9161	3.2497
GS	0.91262	4.4982	665.0173	3.5490
GSA	0.92826	4.1950	587.1322	3.1940
PCA	0.90350	5.1637	710.3275	3.8680
GFPCA	0.89042	4.8472	745.6006	4.0001
CNMF	0.9300	4.4187	591.3190	3.1762
Bayesian Naive	0.95195	3.6428	489.5634	2.6286
Bayesian Sparse	**0.95882**	**3.3345**	**448.1721**	**2.4712**
HySure	0.9465	3.8767	511.8525	2.8181

Original and fused images

[Loncan et al. , 15]

Original and fused images

[Loncan et al. , 15]

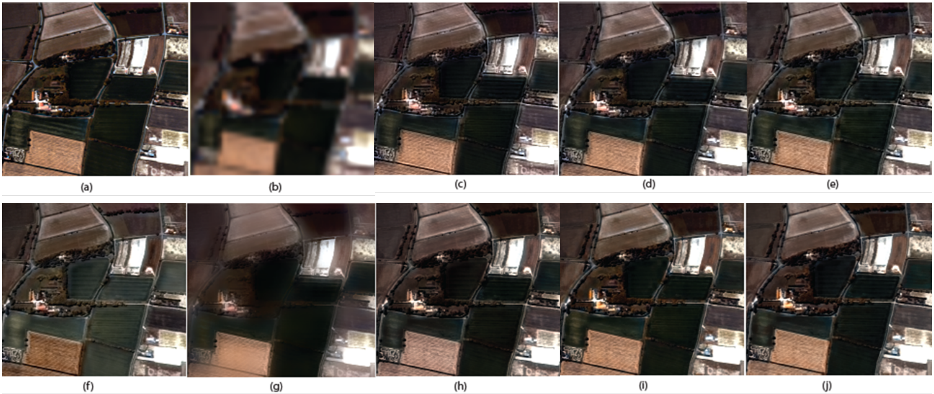


Fig. 6. Details of original and fused Camargue dataset HS image in the visible domain. (a) reference image, (b) interpolated HS image, (c) SFIM, (d) MTF-GLP-HPM, (e) GSA, (f) PCA, (g) GFPCA, (h) CNMF, (i) Bayesian Sparse, (j) HySure

Inverse problems: Plug-and-Play approach

Inverse problems: Plug-and-Play approach

Inverse problems: Plug-and-Play approach

- Variational/MAP criterion (assuming Gaussian noise):

$$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}} \frac{1}{2\,\lambda} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \phi(\mathbf{x})$$

where ϕ is a (hopefully convex) regularizer

Inverse problems: Plug-and-Play approach

- Variational/MAP criterion (assuming Gaussian noise):

$$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}} \frac{1}{2\,\lambda}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \phi(\mathbf{x})$$

  where ϕ is a (hopefully convex) regularizer

- Usually tackled by some iterative algorithm (IST, SpaRSA, TwIST, FISTA, ADMM, DRS, PD, ...). All require the proximity operator of ϕ

$$\text{prox}_{\Phi}(\mathbf{v}) = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - \mathbf{v}\|_2^2 + \phi(\mathbf{x})$$

Inverse problems: Plug-and-Play approach

- Variational/MAP criterion (assuming Gaussian noise):

$$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}} \frac{1}{2\,\lambda}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \phi(\mathbf{x})$$

  where ϕ is a (hopefully convex) regularizer

- Usually tackled by some iterative algorithm (IST, SpaRSA, TwIST, FISTA, ADMM, DRS, PD, ...). All require the proximity operator of ϕ

$$\text{prox}_{\Phi}(\mathbf{v}) = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - \mathbf{v}\|_2^2 + \phi(\mathbf{x})$$

- Clearly, the prox is a denoising operator

Inverse problems: Plug-and-Play approach

- Variational/MAP criterion (assuming Gaussian noise):

$$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}} \frac{1}{2\lambda}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \phi(\mathbf{x})$$

  where ϕ is a (hopefully convex) regularizer

- Usually tackled by some iterative algorithm (IST, SpaRSA, TwIST, FISTA, ADMM, DRS, PD, ...). All require the proximity operator of ϕ

$$\text{prox}_{\Phi}(\mathbf{v}) = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - \mathbf{v}\|_2^2 + \phi(\mathbf{x})$$

- Clearly, the prox is a denoising operator

- Plug-and-play (PnP) approach: replace prox_{ϕ} with a state-of-the-art denoiser [Venkatakrishnan et al., 13]

Inverse problems: Plug-and-Play approach

- Variational/MAP criterion (assuming Gaussian noise):

$$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}} \frac{1}{2\,\lambda}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \phi(\mathbf{x})$$

  where ϕ is a (hopefully convex) regularizer

- Usually tackled by some iterative algorithm (IST, SpaRSA, TwIST, FISTA, ADMM, DRS, PD, ...). All require the proximity operator of ϕ

$$\text{prox}_{\Phi}(\mathbf{v}) = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - \mathbf{v}\|_2^2 + \phi(\mathbf{x})$$

- Clearly, the prox is a denoising operator
- Plug-and-play (PnP) approach: replace prox_{ϕ} with a state-of-the-art denoiser [Venkatakrishnan et al., 13]
- Another strategy: regularization by denoising (ReD) [Romano et al., 16]

PnP-ADMM

- Optimization problem: $\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda\phi(\mathbf{x})$

PnP-ADMM

- Optimization problem: $\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda\phi(\mathbf{x})$
- ADMM directly applied to this problem has the form

$$
\begin{aligned}
\mathbf{x}_{k+1} &= \left(\mathbf{A}^T\mathbf{A} + \rho\mathbf{I}\right)^{-1}\left(\mathbf{A}^T\mathbf{y} + \rho(\mathbf{z}_k + \mathbf{u}_k)\right) \\
\mathbf{z}_{k+1} &= \mathsf{prox}_{\lambda\phi/\rho}\left(\mathbf{x}_{k+1} - \mathbf{u}_k\right) \qquad \text{(denoiser)} \\
\mathbf{u}_{k+1} &= \mathbf{u}_{k+1} - \mathbf{x}_{k+1} + \mathbf{z}_{k+1}
\end{aligned}
$$

PnP-ADMM

- Optimization problem: $\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda\phi(\mathbf{x})$
- ADMM directly applied to this problem has the form

$$\begin{aligned}
\mathbf{x}_{k+1} &= \left(\mathbf{A}^T\mathbf{A} + \rho\mathbf{I}\right)^{-1}\left(\mathbf{A}^T\mathbf{y} + \rho(\mathbf{z}_k + \mathbf{u}_k)\right) \\
\mathbf{z}_{k+1} &= \mathsf{prox}_{\lambda\phi/\rho}\left(\mathbf{x}_{k+1} - \mathbf{u}_k\right) \qquad \text{(denoiser)} \\
\mathbf{u}_{k+1} &= \mathbf{u}_{k+1} - \mathbf{x}_{k+1} + \mathbf{z}_{k+1}
\end{aligned}$$

- Most state-of-the-art denoisers do not have the form of a prox (at least, explicitly)

PnP-ADMM

- Optimization problem: $\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda\phi(\mathbf{x})$
- ADMM directly applied to this problem has the form

$$\mathbf{x}_{k+1} = \left(\mathbf{A}^T\mathbf{A} + \rho\mathbf{I}\right)^{-1}\left(\mathbf{A}^T\mathbf{y} + \rho(\mathbf{z}_k + \mathbf{u}_k)\right)$$
$$\mathbf{z}_{k+1} = \text{prox}_{\lambda\phi/\rho}\left(\mathbf{x}_{k+1} - \mathbf{u}_k\right) \qquad \text{(denoiser)}$$
$$\mathbf{u}_{k+1} = \mathbf{u}_{k+1} - \mathbf{x}_{k+1} + \mathbf{z}_{k+1}$$

- Most state-of-the-art denoisers do not have the form of a prox (at least, explicitly)
- PnP-ADMM: plug a state-of-the-art denoiser instead of the prox:

✓ Collaborative filtering ([Dabov et al., 07])
✓ Non-local Bayes [Lebrun et al., 13]
✓ Deep neural networks [Burger et al., 12, Xie et al., 2012, Zhang et al., 17]
✓ Patch-based GMM-MMSE [Teodoro et al., 15, 16]
- Global Local Factorization [Zhuang, B-D, 17]

Plug-and-Play ADMM

- Plug a black-box denoiser into ADMM [Venkatakrishnan et al., 13

$$\mathbf{x}_{k+1} = \big(\mathbf{A}^T\mathbf{A} + \rho\mathbf{I}\big)^{-1}\big(\mathbf{A}^T\mathbf{y} + \rho(\mathbf{z}_k + \mathbf{u}_k)\big)$$
$$\mathbf{z}_{k+1} = \text{denoiser}\big(\mathbf{x}_{k+1} - \mathbf{u}_k, 1/\rho\big)$$
$$\mathbf{u}_{k+1} = \mathbf{u}_{k+1} - \mathbf{x}_{k+1} + \mathbf{z}_{k+1}$$

where denoiser$(\cdot, \tau)$ assumes noise variance τ

Plug-and-Play ADMM

- Plug a black-box denoiser into ADMM [Venkatakrishnan et al., 13

$$\begin{aligned} \mathbf{x}_{k+1} &= \left(\mathbf{A}^T\mathbf{A} + \rho\mathbf{I}\right)^{-1}\left(\mathbf{A}^T\mathbf{y} + \rho(\mathbf{z}_k + \mathbf{u}_k)\right) \\ \mathbf{z}_{k+1} &= \text{denoiser}\left(\mathbf{x}_{k+1} - \mathbf{u}_k, 1/\rho\right) \\ \mathbf{u}_{k+1} &= \mathbf{u}_{k+1} - \mathbf{x}_{k+1} + \mathbf{z}_{k+1} \end{aligned}$$

  where denoiser$\left(\cdot, \tau\right)$ assumes noise variance τ

- If denoiser $=$ prox$_\phi$, for convex ϕ, convergence is well-known [Eckstein and Bertsekas, 1992, Boyd et al., 11]..

- ...what about convergence of PnP-ADMM?
  [Sreehari et al., 16, Teodoro et al., 17b, Chan et al., 17] More later...

Plug-and-Play ADMM

- Plug a black-box denoiser into ADMM [Venkatakrishnan et al., 13

$$\mathbf{x}_{k+1} = \big(\mathbf{A}^T\mathbf{A} + \rho\mathbf{I}\big)^{-1}\big(\mathbf{A}^T\mathbf{y} + \rho(\mathbf{z}_k + \mathbf{u}_k)\big)$$
$$\mathbf{z}_{k+1} = \text{denoiser}\big(\mathbf{x}_{k+1} - \mathbf{u}_k, 1/\rho\big)$$
$$\mathbf{u}_{k+1} = \mathbf{u}_{k+1} - \mathbf{x}_{k+1} + \mathbf{z}_{k+1}$$

  where denoiser$(\cdot, \tau)$ assumes noise variance τ

- If denoiser $=$ prox$_\phi$, for convex ϕ, convergence is well-known [Eckstein and Bertsekas, 1992, Boyd et al., 11]..

- ...what about convergence of PnP-ADMM? [Sreehari et al., 16, Teodoro et al., 17b, Chan et al., 17] More later...

- Empirical results: competitive!

Plug-and-Play ADMM: Experiments

Plug-and-Play ADMM: Experiments

ISNR (dB)

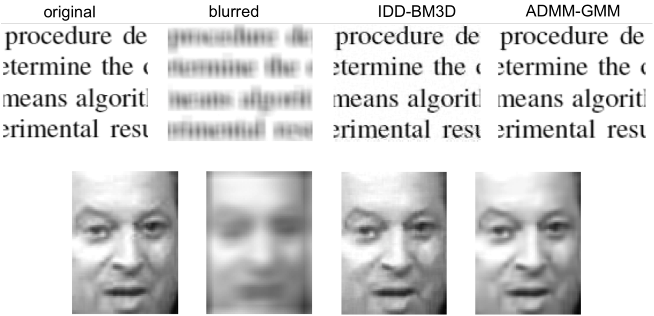
Image:	Cameraman						House					
Experiment:	1	2	3	4	5	6	1	2	3	4	5	6
BSNR	31.87	25.85	40.00	18.53	29.19	17.76	29.16	23.14	40.00	15.99	26.61	15.15
Input PSNR	22.23	22.16	20.76	24.62	23.36	29.82	25.61	25.46	24.11	28.06	27.81	29.98
IDD-BM3D	**8.85**	**7.12**	**10.45**	**3.98**	**4.31**	4.89	**9.95**	**8.55**	12.89	**5.79**	**5.74**	7.13
ADMM-GMM	8.34	6.39	9.73	3.49	4.18	**4.90**	9.84	8.40	12.87	5.57	5.55	6.65
ADMM-BM3D	8.18	6.13	9.58	3.26	3.93	4.88	9.64	8.02	**12.95**	5.23	5.06	**7.37**
Image:	Lena						Barbara					
Experiment:	1	2	3	4	5	6	1	2	3	4	5	6
BSNR	29.89	23.87	40.00	16.47	27.18	15.52	30.81	24.79	40.00	17.35	28.07	16.59
Input PSNR	27.25	27.04	25.84	28.81	29.16	30.03	23.34	23.25	22.49	24.22	23.77	29.78
IDD-BM3D	7.97	**6.61**	8.91	**4.97**	**4.85**	6.34	7.64	**3.96**	**6.05**	1.88	1.16	5.45
ADMM-GMM	**8.01**	6.53	8.95	4.93	4.81	6.09	5.91	2.19	5.37	1.42	1.24	5.14
ADMM-BM3D	8.00	6.56	**9.00**	4.88	4.67	**6.42**	7.32	2.99	**6.05**	1.55	**1.40**	5.76

Class-Adapted GMM-based restoration

- Beating state-of-the-art general-purpose denoisers: divide and conquer, i.e., learn class-adapted denoisers.

Class-Adapted GMM-based restoration

- Beating state-of-the-art general-purpose denoisers: divide and conquer, i.e., learn class-adapted denoisers.
- Learn a GMM for a class of images; use the corresponding patch-based MMSE denoiser [Teodoro et al., 16]



Class-Adapted GMM-based restoration

- Beating state-of-the-art general-purpose denoisers: divide and conquer, i.e., learn class-adapted denoisers.
- Learn a GMM for a class of images; use the corresponding patch-based MMSE denoiser [Teodoro et al., 16]

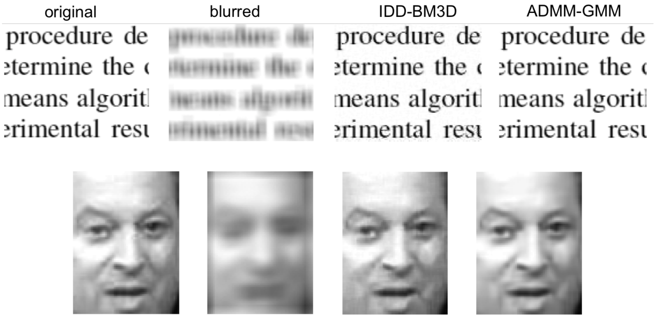


Image class:	Text						Face					
Experiment:	1	2	3	4	5	6	1	2	3	4	5	6
BSNR	26.07	20.05	40.00	15.95	24.78	18.11	28.28	22.26	40.00	15.89	26.22	15.37
Input PSNR	14.14	14.13	12.13	16.83	14.48	28.73	25.61	22.54	20.71	26.49	24.79	30.03
IDD-BM3D	11.97	8.91	16.29	5.88	6.81	4.87	13.66	11.16	14.96	7.31	10.33	6.18
ADMM-GMM	**16.24**	**11.55**	**23.11**	**8.88**	**10.77**	**8.34**	**15.05**	**12.59**	**17.28**	**8.84**	**11.69**	**7.32**

Convergence

- PnP-ADMM with a denoiser

$$
\begin{aligned}
\mathbf{x}_{k+1} &= \left(\mathbf{A}^T\mathbf{A} + \rho\mathbf{I}\right)^{-1}\left(\mathbf{A}^T\mathbf{y} + \rho(\mathbf{z}_k + \mathbf{u}_k)\right) \\
\mathbf{z}_{k+1} &= \text{denoiser}\left(\mathbf{x}_{k+1} - \mathbf{u}_k, 1/\rho\right) \\
\mathbf{u}_{k+1} &= \mathbf{u}_{k+1} - \mathbf{x}_{k+1} + \mathbf{z}_{k+1}
\end{aligned}
$$

Convergence

- PnP-ADMM with a denoiser

$$\begin{aligned}\mathbf{x}_{k+1} &= \left(\mathbf{A}^T\mathbf{A} + \rho\mathbf{I}\right)^{-1}\left(\mathbf{A}^T\mathbf{y} + \rho(\mathbf{z}_k + \mathbf{u}_k)\right)\\ \mathbf{z}_{k+1} &= \text{denoiser}\left(\mathbf{x}_{k+1} - \mathbf{u}_k, 1/\rho\right)\\ \mathbf{u}_{k+1} &= \mathbf{u}_{k+1} - \mathbf{x}_{k+1} + \mathbf{z}_{k+1}\end{aligned}$$

- denoiser is the prox of a convex function $\Rightarrow$ convergence.

Convergence

- PnP-ADMM with a denoiser

$$\begin{aligned}
\mathbf{x}_{k+1} &= \left(\mathbf{A}^T\mathbf{A} + \rho\mathbf{I}\right)^{-1}\left(\mathbf{A}^T\mathbf{y} + \rho(\mathbf{z}_k + \mathbf{u}_k)\right) \\
\mathbf{z}_{k+1} &= \text{denoiser}\left(\mathbf{x}_{k+1} - \mathbf{u}_k, 1/\rho\right) \\
\mathbf{u}_{k+1} &= \mathbf{u}_{k+1} - \mathbf{x}_{k+1} + \mathbf{z}_{k+1}
\end{aligned}$$

- denoiser is the prox of a convex function $\Rightarrow$ convergence.

- From [Moreau 1965]: some map $p : \mathbb{R}^n \to \mathbb{R}^n$ is the prox of a convex function if and only if:
  a) p is non-expansive, *i.e.*, $\forall\, \mathbf{x}, \mathbf{x}',\ \|p(\mathbf{x}) - p(\mathbf{x}')\| \leq \|\mathbf{x} - \mathbf{x}'\|$
  b) and p is subgradient of a convex function, *i.e.*, $\exists\, \phi : \mathbb{R}^n \to \mathbb{R} :\ p(\mathbf{x}) \in \partial\phi(\mathbf{x}),\ \forall\, \mathbf{x}$

Convergence

- PnP-ADMM with a denoiser

$$\begin{aligned}\mathbf{x}_{k+1} &= \left(\mathbf{A}^T\mathbf{A} + \rho\mathbf{I}\right)^{-1}\left(\mathbf{A}^T\mathbf{y} + \rho(\mathbf{z}_k + \mathbf{u}_k)\right)\\ \mathbf{z}_{k+1} &= \text{denoiser}\left(\mathbf{x}_{k+1} - \mathbf{u}_k, 1/\rho\right)\\ \mathbf{u}_{k+1} &= \mathbf{u}_{k+1} - \mathbf{x}_{k+1} + \mathbf{z}_{k+1}\end{aligned}$$

- denoiser is the prox of a convex function $\Rightarrow$ convergence.
- From [Moreau 1965]: some map $p : \mathbb{R}^n \to \mathbb{R}^n$ is the prox of a convex function if and only if:
  - a) p is non-expansive, *i.e.*, $\forall\, \mathbf{x}, \mathbf{x}',\ \|p(\mathbf{x}) - p(\mathbf{x}')\| \leq \|\mathbf{x} - \mathbf{x}'\|$
  - b) and p is subgradient of a convex function, *i.e.*, $\exists\, \phi : \mathbb{R}^n \to \mathbb{R} :\ p(\mathbf{x}) \in \partial\phi(\mathbf{x}),\ \forall\, \mathbf{x}$
- Most state-of-the-art denoisers do no satisfy these conditions

Convergence: GMM-MMSE denoiser

- Is the patch-based GMM-MMSE denoiser non-expansive?

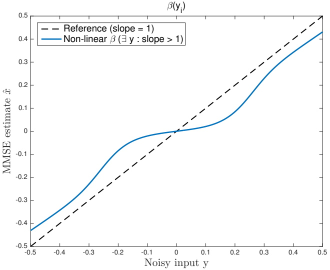
Convergence: GMM-MMSE denoiser

- Is the patch-based GMM-MMSE denoiser non-expansive?
- No! A simple univariate counter-example:
  - ✓ Spike-and-slab-type prior: $p(x) = \frac{1}{2}\mathcal{N}(x; 0, \tau_1) + \frac{1}{2}\mathcal{N}(x; 0, \tau_2), \quad \tau_2 \gg \tau_1$

Convergence: GMM-MMSE denoiser

- Is the patch-based GMM-MMSE denoiser non-expansive?
- No! A simple univariate counter-example:
  - ✓ Spike-and-slab-type prior: $p(x) = \frac{1}{2}\mathcal{N}(x;0,\tau_1) + \frac{1}{2}\mathcal{N}(x;0,\tau_2), \quad \tau_2 \gg \tau_1$
  - ✓ MMSE estimate under Gaussian noise of unit variance:

$$\hat{x} = \mathbb{E}[X|y] = \frac{\frac{\tau_1\, y}{\tau_1+1}\,\beta_1(y) + \frac{\tau_2\, y}{\tau_2+1}\,\beta_2(y)}{\beta_1(y)+\beta_2(y)}, \qquad \text{where } \beta_i(y) = \mathcal{N}(y;0,\tau_i+1)$$

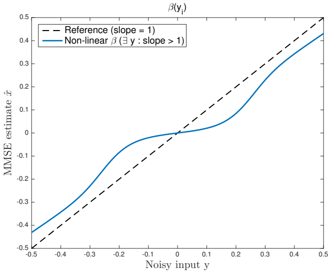


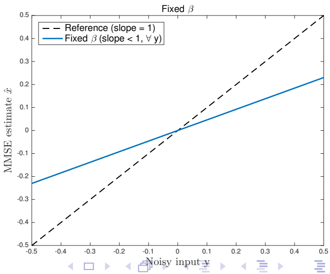
Convergence: GMM-MMSE denoiser

- Is the patch-based GMM-MMSE denoiser non-expansive?
- No! A simple univariate counter-example:
  - ✓ Spike-and-slab-type prior: $p(x) = \frac{1}{2}\mathcal{N}(x; 0, \tau_1) + \frac{1}{2}\mathcal{N}(x; 0, \tau_2), \quad \tau_2 \gg \tau_1$
  - ✓ MMSE estimate under Gaussian noise of unit variance:

$$\hat{x} = \mathbb{E}[X|y] = \frac{\frac{\tau_1\, y}{\tau_1+1}\,\beta_1(y) + \frac{\tau_2\, y}{\tau_2+1}\,\beta_2(y)}{\beta_1(y) + \beta_2(y)}, \qquad \text{where } \beta_i(y) = \mathcal{N}(y; 0, \tau_i + 1)$$

- With β_i fixed: $\hat{x} = y\left(\beta_1 \frac{\tau_1}{\tau_1+1} + \beta_2 \frac{\tau_2}{\tau_2+1}\right)/(\beta_1 + \beta_2)$





Convergence: GMM-MMSE denoiser

- Freeze the weights (β_m) after a certain number of iterations.

Convergence: GMM-MMSE denoiser

- Freeze the weights (β_m) after a certain number of iterations.
- Patch estimate:

$$\hat{\mathbf{x}}_i = \sum_{m=1}^{K} \beta_m^i \, \mathbf{C}_m \left(\mathbf{C}_m + \sigma^2 \, \mathbf{I} \right)^{-1} \mathbf{y}_i$$

Convergence: GMM-MMSE denoiser

- Freeze the weights (β_m) after a certain number of iterations.
- Patch estimate:

$$\hat{\mathbf{x}}_i = \sum_{m=1}^{K} \beta_m^i \, \mathbf{C}_m \left(\mathbf{C}_m + \sigma^2 \, \mathbf{I} \right)^{-1} \mathbf{y}_i = \mathbf{F}_i(\sigma^2) \, \mathbf{y}_i$$

Convergence: GMM-MMSE denoiser

- Freeze the weights (β_m) after a certain number of iterations.
- Patch estimate:

$$\hat{\mathbf{x}}_i = \sum_{m=1}^{K} \beta_m^i \, \mathbf{C}_m \left(\mathbf{C}_m + \sigma^2 \, \mathbf{I} \right)^{-1} \mathbf{y}_i = \mathbf{F}_i(\sigma^2) \, \mathbf{y}_i = \mathbf{F}_i(\sigma^2) \, \mathbf{P}_i \, \mathbf{y}$$

$\mathbf{P}_i$ is the operator (binary matrix) that extracts the i-th patch

Convergence: GMM-MMSE denoiser

- Freeze the weights (β_m) after a certain number of iterations.
- Patch estimate:

$$\hat{\mathbf{x}}_i = \sum_{m=1}^{K} \beta_m^i \, \mathbf{C}_m \left(\mathbf{C}_m + \sigma^2 \, \mathbf{I} \right)^{-1} \mathbf{y}_i = \mathbf{F}_i(\sigma^2) \, \mathbf{y}_i = \mathbf{F}_i(\sigma^2) \, \mathbf{P}_i \, \mathbf{y}$$

$\mathbf{P}_i$ is the operator (binary matrix) that extracts the i-th patch

(weights are normalized, to simplify the notation: $\beta_m^i \leftarrow \beta_m^i / \sum_j \beta_j^i$)

Convergence: GMM-MMSE denoiser

- Freeze the weights (β_m) after a certain number of iterations.
- Patch estimate:

$$\hat{\mathbf{x}}_i = \sum_{m=1}^{K} \beta_m^i \, \mathbf{C}_m \left(\mathbf{C}_m + \sigma^2 \, \mathbf{I} \right)^{-1} \mathbf{y}_i = \mathbf{F}_i(\sigma^2) \, \mathbf{y}_i = \mathbf{F}_i(\sigma^2) \, \mathbf{P}_i \, \mathbf{y}$$

  $\mathbf{P}_i$ is the operator (binary matrix) that extracts the i-th patch

  (weights are normalized, to simplify the notation: $\beta_m^i \leftarrow \beta_m^i / \sum_j \beta_j^i$)

- Global image estimate: aggregate the patch estimates:

$$\hat{\mathbf{x}} = \frac{1}{n_p} \sum_{i=1}^{N} \mathbf{P}_i^T \mathbf{F}_i(\sigma^2) \mathbf{P}_i \, \mathbf{y} = \mathbf{W}(\sigma^2) \, \mathbf{y}$$

Convergence: GMM-MMSE denoiser

- Freeze the weights (β_m) after a certain number of iterations.
- Patch estimate:

$$\hat{\mathbf{x}}_i = \sum_{m=1}^{K} \beta_m^i \, \mathbf{C}_m \Big(\mathbf{C}_m + \sigma^2 \, \mathbf{I} \Big)^{-1} \mathbf{y}_i = \mathbf{F}_i(\sigma^2) \, \mathbf{y}_i = \mathbf{F}_i(\sigma^2) \, \mathbf{P}_i \, \mathbf{y}$$

$\mathbf{P}_i$ is the operator (binary matrix) that extracts the i-th patch

(weights are normalized, to simplify the notation: $\beta_m^i \leftarrow \beta_m^i / \sum_j \beta_j^i$)

- Global image estimate: aggregate the patch estimates:

$$\hat{\mathbf{x}} = \frac{1}{n_p} \sum_{i=1}^{N} \mathbf{P}_i^T \mathbf{F}_i(\sigma^2) \mathbf{P}_i \, \mathbf{y} = \mathbf{W}(\sigma^2) \, \mathbf{y}$$

- Key properties of $\mathbf{W}$ [Teodoro et al., 17c]: for any $\sigma^2 > 0$,

$$\mathbf{W}(\sigma^2) = \mathbf{W}(\sigma^2)^T, \qquad \mathbf{W}(\sigma^2) \succeq 0, \qquad \lambda_{\mathsf{max}}\big(\mathbf{W}(\sigma^2)\big) < 1$$

Convergence: GMM-MMSE denoiser (2)

- Freezing the weights (β_m) after a certain number of iterations,

$$\text{denoiser}(\mathbf{y}, \sigma^2) = \mathbf{W}(\sigma^2)\mathbf{y}$$

Convergence: GMM-MMSE denoiser (2)

- Freezing the weights (β_m) after a certain number of iterations,

$$\text{denoiser}(\mathbf{y}, \sigma^2) = \mathbf{W}(\sigma^2)\mathbf{y}$$

- Recalling Moreau's corollary, this is a proximity operator:
  - It is non-expansive: $\mathbf{W}(\sigma^2)$ is symmetric with $\lambda_{\mathsf{max}}\big(\mathbf{W}(\sigma^2)\big) < 1$
  - It is the gradient of a convex function: $\mathbf{W}(\sigma^2)\mathbf{y} = \nabla_{\mathbf{y}}\big(\frac{1}{2}\mathbf{y}^T\mathbf{W}(\sigma^2)\mathbf{y}\big)$

Convergence: GMM-MMSE denoiser (2)

- Freezing the weights (β_m) after a certain number of iterations,

$$\text{denoiser}(\mathbf{y}, \sigma^2) = \mathbf{W}(\sigma^2)\mathbf{y}$$

- Recalling Moreau's corollary, this is a proximity operator:
  - It is non-expansive: $\mathbf{W}(\sigma^2)$ is symmetric with $\lambda_{\text{max}}\big(\mathbf{W}(\sigma^2)\big) < 1$
  - It is the gradient of a convex function: $\mathbf{W}(\sigma^2)\mathbf{y} = \nabla_{\mathbf{y}}\big(\frac{1}{2}\mathbf{y}^T\mathbf{W}(\sigma^2)\mathbf{y}\big)$
- Corollary:

frozen weights $\Rightarrow$ PnP-ADMM converges

Hyperspectral Fusion via PnP-ADMM

- Assuming Gaussian noise:

$$\widehat{\mathbf{X}} \in \arg\min_{\mathbf{X}\in\mathbb{R}^{p\times n_h}} \frac{1}{2}\|\mathbf{EXBM}-\mathbf{Y}_h\|_F^2 + \frac{\lambda_m}{2}\|\mathbf{REX}-\mathbf{Y}_m\|_F^2 + \text{``}\phi(\mathbf{X})\text{''}$$

Hyperspectral Fusion via PnP-ADMM

- Assuming Gaussian noise:

$$\widehat{\mathbf{X}} \in \arg \min_{\mathbf{X} \in \mathbb{R}^{p \times n_h}} \frac{1}{2} \|\mathbf{EXBM} - \mathbf{Y}_h\|_F^2 + \frac{\lambda_m}{2} \|\mathbf{REX} - \mathbf{Y}_m\|_F^2 + \text{``}\phi(\mathbf{X})\text{''}$$

- We use an instance of ADMM: SALSA [Afonso et al., 11]

Hyperspectral Fusion via PnP-ADMM

- Assuming Gaussian noise:

$$\widehat{\mathbf{X}} \in \arg \min_{\mathbf{X} \in \mathbb{R}^{p \times n_h}} \frac{1}{2} \|\mathbf{EXBM} - \mathbf{Y}_h\|_F^2 + \frac{\lambda_m}{2} \|\mathbf{REX} - \mathbf{Y}_m\|_F^2 + \text{``}\phi(\mathbf{X})\text{''}$$

- We use an instance of ADMM: SALSA [Afonso et al., 11]
- PnP: instead of prox_ϕ, use patch-based GMM denoiser

Hyperspectral Fusion via PnP-ADMM

- Assuming Gaussian noise:

$$\widehat{\mathbf{X}} \in \arg \min_{\mathbf{X} \in \mathbb{R}^{p \times n_h}} \frac{1}{2} \|\mathbf{EXBM} - \mathbf{Y}_h\|_F^2 + \frac{\lambda_m}{2} \|\mathbf{REX} - \mathbf{Y}_m\|_F^2 + \text{“}\phi(\mathbf{X})\text{”}$$

- We use an instance of ADMM: SALSA [Afonso et al., 11]
- PnP: instead of prox_ϕ, use patch-based GMM denoiser
- The GMM is learned from patches of $\mathbf{Y}_m$ (high spatial resolution)

Hyperspectral Fusion via PnP-ADMM

- Assuming Gaussian noise:

$$\widehat{\mathbf{X}} \in \arg\min_{\mathbf{X}\in\mathbb{R}^{p\times n_h}} \frac{1}{2}\|\mathbf{EXBM}-\mathbf{Y}_h\|_F^2 + \frac{\lambda_m}{2}\|\mathbf{REX}-\mathbf{Y}_m\|_F^2 + \text{``}\phi(\mathbf{X})\text{''}$$

- We use an instance of ADMM: SALSA [Afonso et al., 11]
- PnP: instead of prox_ϕ, use patch-based GMM denoiser
- The GMM is learned from patches of $\mathbf{Y}_m$ (high spatial resolution)
- The weights (β_m) are kept fixed: scene adaptation

Hyperspectral Fusion via PnP-ADMM

- Assuming Gaussian noise:

$$\widehat{\mathbf{X}} \in \arg \min_{\mathbf{X} \in \mathbb{R}^{p \times n_h}} \frac{1}{2}\|\mathbf{EXBM} - \mathbf{Y}_h\|_F^2 + \frac{\lambda_m}{2}\|\mathbf{REX} - \mathbf{Y}_m\|_F^2 + \text{``}\phi(\mathbf{X})\text{''}$$

- We use an instance of ADMM: SALSA [Afonso et al., 11]
- PnP: instead of prox_ϕ, use patch-based GMM denoiser
- The GMM is learned from patches of $\mathbf{Y}_m$ (high spatial resolution)
- The weights (β_m) are kept fixed: scene adaptation
- $\Rightarrow$ convergence [Teodoro et al., 17a]

ADMM/SALSA

- Variable splitting reformulation

$$\widehat{\mathbf{X}}, \widehat{\mathbf{V}}_1, \widehat{\mathbf{V}}_2, \widehat{\mathbf{V}}_3 \in \underset{\mathbf{X},\mathbf{V}_1,\mathbf{V}_2,\mathbf{V}_3}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{E}\mathbf{V}_1\mathbf{M} - \mathbf{Y}_h\|_F^2 + \frac{\lambda_m}{2}\|\mathbf{R}\mathbf{E}\mathbf{V}_2 - \mathbf{Y}_m\|_F^2 + \lambda_\phi \phi(\mathbf{V}_3)$$

$$\text{subject to} \quad \mathbf{V}_1 = \mathbf{X}\mathbf{B}, \quad \mathbf{V}_2 = \mathbf{X}, \quad \mathbf{V}_3 = \mathbf{X}$$

ADMM/SALSA

- Variable splitting reformulation

$$\widehat{\mathbf{X}}, \widehat{\mathbf{V}}_1, \widehat{\mathbf{V}}_2, \widehat{\mathbf{V}}_3 \in \underset{\mathbf{X},\mathbf{V}_1,\mathbf{V}_2,\mathbf{V}_3}{\text{argmin}} \ \tfrac{1}{2}\|\mathbf{E}\mathbf{V}_1\mathbf{M} - \mathbf{Y}_h\|_F^2 + \tfrac{\lambda_m}{2}\|\mathbf{R}\mathbf{E}\mathbf{V}_2 - \mathbf{Y}_m\|_F^2 + \lambda_\phi \phi(\mathbf{V}_3)$$

$$\text{subject to} \quad \mathbf{V}_1 = \mathbf{X}\mathbf{B}, \quad \mathbf{V}_2 = \mathbf{X}, \quad \mathbf{V}_3 = \mathbf{X}$$

- SALSA/ADMM: solve sequence of simpler sub-problems (*e.g.* using ADMM)

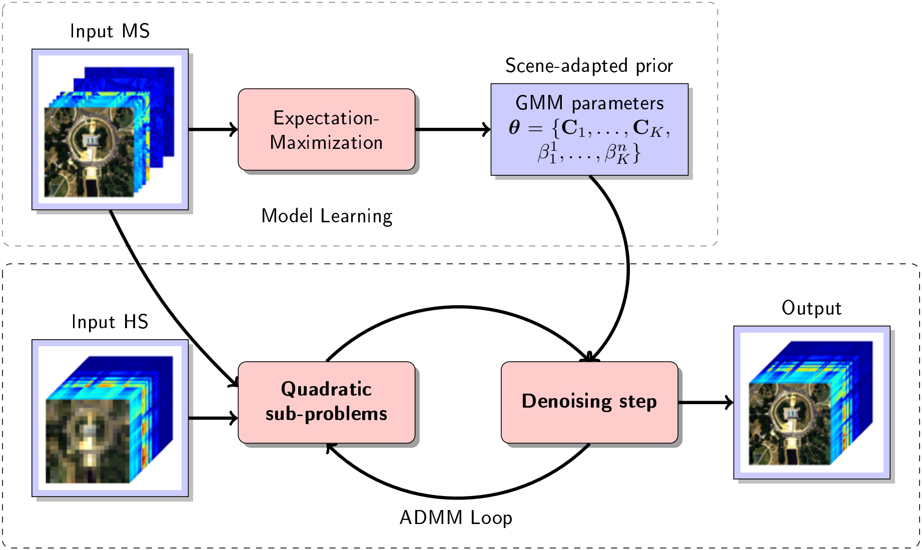
$$\mathbf{X}^{k+1} = \underset{\mathbf{X}}{\text{argmin}} \ \|\mathbf{X}\mathbf{B}-\mathbf{V}_1-\mathbf{D}_1\|_F^2 + \|\mathbf{X}-\mathbf{V}_2-\mathbf{D}_2\|_F^2 + \|\mathbf{X}-\mathbf{V}_3-\mathbf{D}_3\|_F^2,$$

$$\mathbf{V}_1^{k+1} = \underset{\mathbf{V}_1}{\text{argmin}} \ \frac{1}{2}\|\mathbf{E}\mathbf{V}_1\mathbf{M} - \mathbf{Y}_h\|_F^2 + \frac{\rho}{2}\|\mathbf{X}^{k+1}\mathbf{B} - \mathbf{V}_1 - \mathbf{D}_1^k\|_F^2,$$

$$\mathbf{V}_2^{k+1} = \underset{\mathbf{V}_2}{\text{argmin}} \ \frac{\lambda_m}{2}\|\mathbf{R}\mathbf{E}\mathbf{V}_2 - \mathbf{Y}_m\|_F^2 + \frac{\rho}{2}\|\mathbf{X}^{k+1} - \mathbf{V}_2 - \mathbf{D}_2^k\|_F^2,$$

$$\mathbf{V}_3^{k+1} = \text{denoise}(\mathbf{X}^{k+1} - \mathbf{D}_3^k, \frac{\lambda_\phi}{\rho})$$

Algorithm



Input MS
Expectation-Maximization
Scene-adapted prior
GMM parameters
$\boldsymbol{\theta} = \{\mathbf{C}_1, \ldots, \mathbf{C}_K, \beta_1^1, \ldots, \beta_K^n\}$
Model Learning
Input HS
Quadratic sub-problems
Denoising step
Output
ADMM Loop

Hyperspectral Fusion: Synthetic Example

Hyperspectral Fusion: Synthetic Example

		Exp. 1 (PAN)			Exp. 2 (PAN)			Exp. 3 (R,G,B,N-IR)			Exp. 4 (R,G,B,N-IR)		
Dataset	Metric	ERGAS	SAM	SRE	ERGAS	SAM	SRE	ERGAS	SAM	SRE	ERGAS	SAM	SRE
Rosis	Dictionary	1.99	3.28	22.64	2.05	3.16	22.32	**0.47**	**0.85**	**34.60**	0.85	1.47	29.66
	GMM	1.75	2.89	23.67	1.92	2.92	22.85	0.48	0.87	34.32	0.91	1.65	29.05
	Modified GMM	**1.65**	**2.75**	**24.17**	**1.81**	**2.76**	**23.31**	0.49	0.87	34.59	**0.80**	**1.42**	**30.14**
Moffett	Dictionaries	2.67	4.18	20.28	2.74	4.20	20.05	1.85	2.72	23.58	2.12	3.21	22.25
	GMM	2.66	4.24	20.26	2.78	4.27	19.87	1.81	2.68	23.81	2.30	3.37	21.63
	Modified GMM	**2.54**	**4.06**	**20.66**	**2.65**	**4.10**	**20.28**	**1.73**	**2.58**	**24.18**	**1.97**	**2.90**	**22.94**

- Metrics: ERGAS = *erreur relative globale adimensionnelle de synthèse*
  SAM = spectral angle mapper (low is good)
  SRE = signal to reconstruction error (dB, high is good)

Spectral Prior

- Leverage spectral information, as well as spatial

Spectral Prior

- Leverage spectral information, as well as spatial
- Dual regularization approach

$$\widehat{\mathbf{X}} \in \underset{\mathbf{X}}{\text{argmin}} \quad \frac{1}{2}\|\mathbf{EXBM} - \mathbf{Y}_h\|_F^2 + \frac{\lambda_m}{2}\|\mathbf{REX} - \mathbf{Y}_m\|_F^2 + \\ + \lambda_{\text{spatial}}\, \phi_{\text{spatial}}(\mathbf{X}) + \lambda_{\text{spectral}}\, \phi_{\text{spectral}}(\mathbf{X}^T)$$

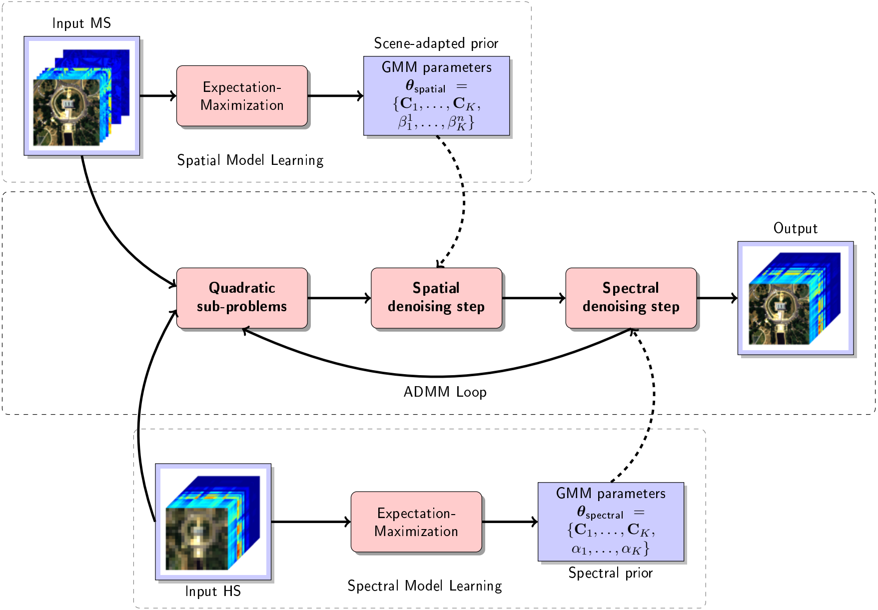
Spectral Prior

- Leverage spectral information, as well as spatial
- Dual regularization approach

$$\widehat{\mathbf{X}} \in \underset{\mathbf{X}}{\text{argmin}} \quad \frac{1}{2}\|\mathbf{EXBM} - \mathbf{Y}_h\|_F^2 + \frac{\lambda_m}{2}\|\mathbf{REX} - \mathbf{Y}_m\|_F^2 + \\ + \lambda_{\text{spatial}}\, \phi_{\text{spatial}}(\mathbf{X}) + \lambda_{\text{spectral}}\, \phi_{\text{spectral}}(\mathbf{X}^T)$$

- Plug-and-play with two denoisers
  - ✓ A GMM spatial prior, learned from the MS image(s)
  - ✓ A GMM spectral prior learned from HS bands, *i.e.*, spectra of each HS pixel

Algorithm


Input MS
Expectation-Maximization
Scene-adapted prior
GMM parameters
$\boldsymbol{\theta}_{\text{spatial}} = \{\mathbf{C}_1, \ldots, \mathbf{C}_K, \beta_1^1, \ldots, \beta_K^n\}$
Spatial Model Learning
Output
Quadratic sub-problems
Spatial denoising step
Spectral denoising step
ADMM Loop
Input HS
Expectation-Maximization
GMM parameters
$\boldsymbol{\theta}_{\text{spectral}} = \{\mathbf{C}_1, \ldots, \mathbf{C}_K, \alpha_1, \ldots, \alpha_K\}$
Spectral prior
Spectral Model Learning

Improving Hyperspectral Fusion Results

- Orthonormal subspace obtained with SVD

Method	Spatial			Spectral			Spatial-Spectral		
Dataset	PSNR	SAM	ERGAS	PSNR	SAM	ERGAS	PSNR	SAM	ERGAS
AVIRIS Indian Pines	*43.10*	**0.60**	**0.31**	**43.22**	*0.61*	**0.31**	42.77	*0.61*	*0.32*
AVIRIS Cuprite	*43.97*	*0.50*	*0.23*	42.66	0.59	0.28	**44.09**	**0.48**	**0.23**
AVIRIS Moffett Field	*34.67*	1.95	*5.23*	34.38	*1.85*	5.28	**36.28**	**1.50**	**4.49**
HYDICE W. DC Mall	*34.59*	3.44	*4.66*	34.19	*1.75*	6.20	**37.16**	**1.53**	**3.94**
HyperSpec Chikusei	*44.70*	*1.49*	*1.53*	41.34	1.70	1.98	**44.76**	**1.39**	**1.48**
ROSIS-3 Univ. Pavia	*39.35*	4.49	*1.29*	37.99	*3.20*	1.34	**40.84**	**3.01**	**0.97**
CASI Univ. Houston	**44.86**	2.17	*1.65*	41.32	*2.01*	2.16	*44.65*	**1.89**	**1.59**
Average	*40.75*	2.09	*2.13*	39.30	*1.67*	2.51	**41.51**	**1.49**	**1.86**

Improving Hyperspectral Fusion Results

- Orthonormal subspace obtained with SVD

Method	Spatial			Spectral			Spatial-Spectral		
Dataset	PSNR	SAM	ERGAS	PSNR	SAM	ERGAS	PSNR	SAM	ERGAS
AVIRIS Indian Pines	*43.10*	**0.60**	**0.31**	**43.22**	*0.61*	**0.31**	42.77	*0.61*	*0.32*
AVIRIS Cuprite	*43.97*	*0.50*	*0.23*	42.66	0.59	0.28	**44.09**	**0.48**	**0.23**
AVIRIS Moffett Field	*34.67*	1.95	*5.23*	34.38	*1.85*	5.28	**36.28**	**1.50**	**4.49**
HYDICE W. DC Mall	*34.59*	3.44	*4.66*	34.19	*1.75*	6.20	**37.16**	**1.53**	**3.94**
HyperSpec Chikusei	*44.70*	*1.49*	*1.53*	41.34	1.70	1.98	**44.76**	**1.39**	**1.48**
ROSIS-3 Univ. Pavia	*39.35*	4.49	*1.29*	37.99	*3.20*	1.34	**40.84**	**3.01**	**0.97**
CASI Univ. Houston	**44.86**	2.17	*1.65*	41.32	*2.01*	2.16	*44.65*	**1.89**	**1.59**
Average	*40.75*	2.09	*2.13*	39.30	*1.67*	2.51	**41.51**	**1.49**	**1.86**

- Subspace obtained with HySime [B-D, Nascimento, 08]

Method	Spatial			Spectral			Spatial-Spectral		
Dataset	PSNR	SAM	ERGAS	PSNR	SAM	ERGAS	PSNR	SAM	ERGAS
AVIRIS Indian Pines	**42.52**	**0.59**	**0.32**	41.70	0.65	0.35	*42.34*	*0.60*	*0.33*
AVIRIS Cuprite	*43.91*	**0.48**	**0.22**	42.83	0.55	*0.25*	**44.38**	**0.48**	**0.22**
AVIRIS Moffett Field	*36.14*	**1.62**	*4.66*	35.92	1.89	4.70	**36.28**	**1.62**	**4.65**
HYDICE W. DC Mall	*37.91*	*1.83*	**3.60**	37.82	1.87	3.62	**37.95**	**1.80**	*3.65*
HyperSpec Chikusei	**46.06**	**1.22**	**1.56**	40.89	1.37	1.63	*46.05*	*1.22*	*1.57*
ROSIS-3 Univ. Pavia	*41.75*	**2.70**	**0.80**	38.47	2.73	*0.83*	**42.71**	**2.70**	**0.80**
CASI Univ. Houston	**47.29**	1.47	*1.13*	43.49	1.47	1.14	*47.28*	**1.45**	**1.12**
Average	*42.23*	*1.42*	**1.76**	40.16	1.50	1.79	**42.43**	**1.41**	**1.76**

Final Remarks

Final Remarks

- Image patches: effective low-dimensional image representation

Final Remarks

- Image patches: effective low-dimensional image representation
- GMM for MMSE patch estimation: a flexible tool/model

Final Remarks

- Image patches: effective low-dimensional image representation
- GMM for MMSE patch estimation: a flexible tool/model
- Specialize to image classes

Final Remarks

- Image patches: effective low-dimensional image representation
- GMM for MMSE patch estimation: a flexible tool/model
- Specialize to image classes
- Tighter adaptation: scene-adapted for hyperspectral denoising, destripping, superresolution, fusion, compressive sensing
- Tool: plug-and-play ADMM

Final Remarks

- Image patches: effective low-dimensional image representation
- GMM for MMSE patch estimation: a flexible tool/model
- Specialize to image classes
- Tighter adaptation: scene-adapted for hyperspectral denoising, destripping, superresolution, fusion, compressive sensing
- Tool: plug-and-play ADMM
- Convergence guarantee (GMM with fixed weights, BM3D with fixed groups)

Final Remarks

- Image patches: effective low-dimensional image representation
- GMM for MMSE patch estimation: a flexible tool/model
- Specialize to image classes
- Tighter adaptation: scene-adapted for hyperspectral denoising, destripping, superresolution, fusion, compressive sensing
- Tool: plug-and-play ADMM
- Convergence guarantee (GMM with fixed weights, BM3D with fixed groups)
- Ongoing work: non-Gaussian noise

Final Remarks

- Image patches: effective low-dimensional image representation
- GMM for MMSE patch estimation: a flexible tool/model
- Specialize to image classes
- Tighter adaptation: scene-adapted for hyperspectral denoising, destripping, superresolution, fusion, compressive sensing
- Tool: plug-and-play ADMM
- Convergence guarantee (GMM with fixed weights, BM3D with fixed groups)
- Ongoing work: non-Gaussian noise
- Ongoing work: PnP with other algorithms (DNNs)

Current research