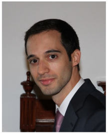
Alternating Direction Method of Multipliers in Imaging: Overview of a Line of Work

Mário A. T. Figueiredo

Joint work with **José Bioucas-Dias**, Manya Afonso, Mariana Almeida, Afonso Teodoro, Marina Ljubenovic, ...

1960-2020



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

ADMM, Edinburgh, 2022

Outline

Outline

Alternating Direction Methof of Multipliers (ADMM)

- Canonical problem:

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n,\, \mathbf{z}\in\mathbb{R}^m} \quad & f(\mathbf{x}) + g(\mathbf{z}) \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{b} \end{aligned}$$

Alternating Direction Methof of Multipliers (ADMM)

- Canonical problem:

$$\begin{array}{ll} \displaystyle\min_{\mathbf{x}\in\mathbb{R}^n,\, \mathbf{z}\in\mathbb{R}^m} & f(\mathbf{x}) + g(\mathbf{z}) \\ \text{subject to} & \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{b} \end{array}$$

- Functions $f : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$ and $g : \mathbb{R}^m \rightarrow \bar{\mathbb{R}}$ are closed, proper, and convex

Alternating Direction Methof of Multipliers (ADMM)

- Canonical problem:

$$\begin{array}{rc} \displaystyle\min_{\mathbf{x}\in\mathbb{R}^n,\, \mathbf{z}\in\mathbb{R}^m} & f(\mathbf{x}) + g(\mathbf{z}) \\ \text{subject to} & \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{b} \end{array}$$

- Functions $f : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$ and $g : \mathbb{R}^m \rightarrow \bar{\mathbb{R}}$ are closed, proper, and convex

- Often used to re-write problems of the form

$$\min_{\mathbf{x}} \; f(\mathbf{x}) + g(\mathbf{H}\mathbf{x})$$

as

$$\min_{\mathbf{x},\mathbf{z}} \; f(\mathbf{x}) + g(\mathbf{z}) \quad \text{subject to} \quad \mathbf{H}\mathbf{x} = \mathbf{z}$$

Alternating Direction Method of Multipliers (ADMM)

- Canonical problem:

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n,\, \mathbf{z}\in\mathbb{R}^m} \quad & f(\mathbf{x}) + g(\mathbf{z}) \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{b} \end{aligned}$$

Alternating Direction Method of Multipliers (ADMM)

- Canonical problem:
$$\min_{\mathbf{x}\in\mathbb{R}^n,\, \mathbf{z}\in\mathbb{R}^m} \quad f(\mathbf{x}) + g(\mathbf{z})$$
$$\text{subject to} \quad \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{b}$$

- Canonical ADMM (in scaled form)

$$\begin{aligned}
\mathbf{x}_{k+1} &= \arg\min_{\mathbf{x}} f(\mathbf{x}) + \frac{\rho}{2}\big\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z}_k - \mathbf{b} + \mathbf{u}_k\big\|_2^2 \\
\mathbf{z}_{k+1} &= \arg\min_{\mathbf{z}} g(\mathbf{z}) + \frac{\rho}{2}\big\|\mathbf{A}\mathbf{x}_{k+1} + \mathbf{B}\mathbf{z} - \mathbf{b} + \mathbf{u}_k\big\|_2^2 \\
\mathbf{u}_{k+1} &= \mathbf{u}_{k+1} + \mathbf{A}\mathbf{x}_{k+1} + \mathbf{B}\mathbf{z}_{k+1} - \mathbf{b}
\end{aligned}$$

Alternating Direction Method of Multipliers (ADMM)

- Canonical problem: $$\min_{\mathbf{x}\in\mathbb{R}^n,\ \mathbf{z}\in\mathbb{R}^m} \quad f(\mathbf{x}) + g(\mathbf{z})$$ $$\text{subject to} \quad \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{b}$$

- Canonical ADMM (in scaled form)

$$\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}} f(\mathbf{x}) + \frac{\rho}{2}\left\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z}_k - \mathbf{b} + \mathbf{u}_k\right\|_2^2$$
$$\mathbf{z}_{k+1} = \arg\min_{\mathbf{z}} g(\mathbf{z}) + \frac{\rho}{2}\left\|\mathbf{A}\mathbf{x}_{k+1} + \mathbf{B}\mathbf{z} - \mathbf{b} + \mathbf{u}_k\right\|_2^2$$
$$\mathbf{u}_{k+1} = \mathbf{u}_{k+1} + \mathbf{A}\mathbf{x}_{k+1} + \mathbf{B}\mathbf{z}_{k+1} - \mathbf{b}$$

- Can be derived in several ways: method of multipliers (augmented Lagrangian); Douglas-Rachford for the dual; ...

Alternating Direction Method of Multipliers (ADMM)

- Canonical problem:
$$\min_{\mathbf{x}\in\mathbb{R}^n,\ \mathbf{z}\in\mathbb{R}^m} \quad f(\mathbf{x}) + g(\mathbf{z})$$
$$\text{subject to} \quad \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{b}$$

- Canonical ADMM (in scaled form)
$$\begin{aligned}
\mathbf{x}_{k+1} &= \arg\min_{\mathbf{x}} f(\mathbf{x}) + \frac{\rho}{2}\big\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z}_k - \mathbf{b} + \mathbf{u}_k\big\|_2^2 \\
\mathbf{z}_{k+1} &= \arg\min_{\mathbf{z}} g(\mathbf{z}) + \frac{\rho}{2}\big\|\mathbf{A}\mathbf{x}_{k+1} + \mathbf{B}\mathbf{z} - \mathbf{b} + \mathbf{u}_k\big\|_2^2 \\
\mathbf{u}_{k+1} &= \mathbf{u}_{k+1} + \mathbf{A}\mathbf{x}_{k+1} + \mathbf{B}\mathbf{z}_{k+1} - \mathbf{b}
\end{aligned}$$

- Can be derived in several ways: method of multipliers (augmented Lagrangian); Douglas-Rachford for the dual; ...

- Introduced by French mathematicians in the 1970s
[Gabay and Mercier, 1976], [Glowinski and Marrocco, 1975]

Alternating Direction Method of Multipliers (ADMM)

- Canonical problem:
$$\min_{\mathbf{x}\in\mathbb{R}^n,\ \mathbf{z}\in\mathbb{R}^m} \quad f(\mathbf{x}) + g(\mathbf{z})$$
$$\text{subject to} \quad \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{b}$$

- Canonical ADMM (in scaled form)
$$\begin{aligned}
\mathbf{x}_{k+1} &= \arg\min_{\mathbf{x}} f(\mathbf{x}) + \frac{\rho}{2}\big\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z}_k - \mathbf{b} + \mathbf{u}_k\big\|_2^2\\
\mathbf{z}_{k+1} &= \arg\min_{\mathbf{z}} g(\mathbf{z}) + \frac{\rho}{2}\big\|\mathbf{A}\mathbf{x}_{k+1} + \mathbf{B}\mathbf{z} - \mathbf{b} + \mathbf{u}_k\big\|_2^2\\
\mathbf{u}_{k+1} &= \mathbf{u}_{k+1} + \mathbf{A}\mathbf{x}_{k+1} + \mathbf{B}\mathbf{z}_{k+1} - \mathbf{b}
\end{aligned}$$

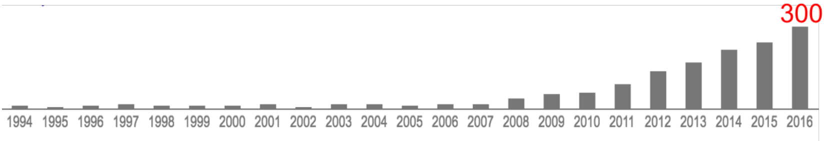
- Can be derived in several ways: method of multipliers (augmented Lagrangian); Douglas-Rachford for the dual; ...

- Introduced by French mathematicians in the 1970s [Gabay and Mercier, 1976], [Glowinski and Marrocco, 1975]

- *Cornerstone* work in the 1990s by Eckstein and Bertsekas [1992]

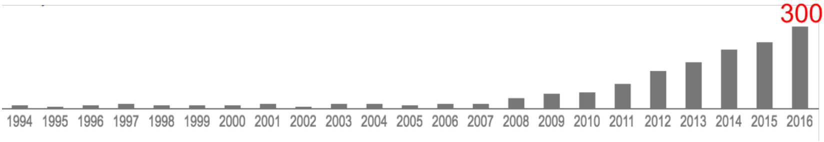
Explosion of Interest in ADMM

- Citations to paper by Eckstein and Bertsekas [1992]:



Explosion of Interest in ADMM

- Citations to paper by Eckstein and Bertsekas [1992]:



- Citations to review paper by Boyd et al. [2011]:

Classical Convergence Result

- Problem: $\min_{\mathbf{x}} \; f(\mathbf{x}) + g(\mathbf{H}\mathbf{x})$

- ADMM:

$$\mathbf{x}^{(k+1)} = \arg\min_{\mathbf{x}} f(\mathbf{x}) + \frac{\rho}{2}\big\|\mathbf{H}\mathbf{x} - \mathbf{v}^{(k)} - \mathbf{u}^{(k)}\big\|_2^2$$
$$\mathbf{v}^{(k+1)} = \arg\min_{\mathbf{v}} g(\mathbf{v}) + \frac{\rho}{2}\big\|\mathbf{H}\mathbf{x}^{(k+1)} - \mathbf{v} - \mathbf{u}^{(k)}\big\|_2^2,$$
$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} - \mathbf{H}\mathbf{x}^{(k+1)} + \mathbf{v}^{(k+1)},$$

Classical Convergence Result

- Problem: $\min_{\mathbf{x}} \; f(\mathbf{x}) + g(\mathbf{H}\mathbf{x})$

- ADMM:

$$\mathbf{x}^{(k+1)} = \arg\min_{\mathbf{x}} f(\mathbf{x}) + \frac{\rho}{2}\big\|\mathbf{H}\mathbf{x} - \mathbf{v}^{(k)} - \mathbf{u}^{(k)}\big\|_2^2$$
$$\mathbf{v}^{(k+1)} = \arg\min_{\mathbf{v}} g(\mathbf{v}) + \frac{\rho}{2}\big\|\mathbf{H}\mathbf{x}^{(k+1)} - \mathbf{v} - \mathbf{u}^{(k)}\big\|_2^2,$$
$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} - \mathbf{H}\mathbf{x}^{(k+1)} + \mathbf{v}^{(k+1)},$$

Theorem (Eckstein and Bertsekas [1992] (simplified version))

Let $\mathbf{H}$ have full column rank, and $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ and $g : \mathbb{R}^m \to \bar{\mathbb{R}}$ be closed, proper, convex functions; let $\mathbf{v}_0, \mathbf{u}_0 \in \mathbb{R}^m$, and $\rho > 0$ be given. Then $(\mathbf{x}^{(k)})_{k=1,2,\ldots}$ converges to a solution, if one exists. If no solution exists, then at least one of the sequences $(\mathbf{v}^{(k)})_{k=1,2,\ldots}$ or $(\mathbf{u}^{(k)})_{k=1,2,\ldots}$ diverges.

Classical Convergence Result

- Problem: $\min_{\mathbf{x}} \; f(\mathbf{x}) + g(\mathbf{Hx})$

- ADMM:

$$\mathbf{x}^{(k+1)} = \arg\min_{\mathbf{x}} f(\mathbf{x}) + \frac{\rho}{2}\big\|\mathbf{Hx} - \mathbf{v}^{(k)} - \mathbf{u}^{(k)}\big\|_2^2 \qquad \text{(not a prox)}$$

$$\mathbf{v}^{(k+1)} = \arg\min_{\mathbf{v}} g(\mathbf{v}) + \frac{\rho}{2}\big\|\mathbf{Hx}^{(k+1)} - \mathbf{v} - \mathbf{u}^{(k)}\big\|_2^2, \qquad \text{(a prox)}$$

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} - \mathbf{Hx}^{(k+1)} + \mathbf{v}^{(k+1)},$$

Theorem (Eckstein and Bertsekas [1992] (simplified version))

Let $\mathbf{H}$ have full column rank, and $f : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$ and $g : \mathbb{R}^m \rightarrow \bar{\mathbb{R}}$ be closed, proper, convex functions; let $\mathbf{v}_0, \mathbf{u}_0 \in \mathbb{R}^m$, and $\rho > 0$ be given. Then $(\mathbf{x}^{(k)})_{k=1,2,\ldots}$ converges to a solution, if one exists. If no solution exists, then at least one of the sequences $(\mathbf{v}^{(k)})_{k=1,2,\ldots}$ or $(\mathbf{u}^{(k)})_{k=1,2,\ldots}$ diverges.

- **Prox**imity operator: $\text{prox}_\phi(\mathbf{u}) := \arg\min_{\mathbf{x}} \; \phi(\mathbf{x}) + \frac{1}{2}\|\mathbf{u} - \mathbf{x}\|_2^2$

  Moreau [1965]

Two or More Functions

- Problem template: $$\min_{\mathbf{x}\in\mathbb{R}^n} \sum_{j=1}^{J} g_j\big(\mathbf{H}_j\,\mathbf{x}\big)$$

Two or More Functions

- Problem template: $$\min_{\mathbf{x}\in\mathbb{R}^n} \sum_{j=1}^{J} g_j\big(\mathbf{H}_j\,\mathbf{x}\big)$$
  - ✓ $g_j : \mathbb{R}^{m_j} \rightarrow \bar{\mathbb{R}}$ are closed, proper, and convex.

Two or More Functions

- Problem template: $$\min_{\mathbf{x}\in\mathbb{R}^n} \sum_{j=1}^{J} g_j\big(\mathbf{H}_j\,\mathbf{x}\big)$$
  - ✓ $g_j : \mathbb{R}^{m_j} \rightarrow \bar{\mathbb{R}}$ are closed, proper, and convex.
  - ✓ $\mathbf{H}_j \in \mathbb{R}^{m_j \times n}$

Two or More Functions

- Problem template: $$\min_{\mathbf{x}\in\mathbb{R}^n} \sum_{j=1}^{J} g_j\left(\mathbf{H}_j\,\mathbf{x}\right)$$
  - ✓ $g_j : \mathbb{R}^{m_j} \to \bar{\mathbb{R}}$ are closed, proper, and convex.
  - ✓ $\mathbf{H}_j \in \mathbb{R}^{m_j \times n}$
- Can be re-written in canonical form

$$\min_{\mathbf{x}} \; f(\mathbf{x}) + g(\mathbf{H}\mathbf{x}),$$

with

Two or More Functions

- Problem template: $$\min_{\mathbf{x}\in\mathbb{R}^n} \sum_{j=1}^{J} g_j\big(\mathbf{H}_j\,\mathbf{x}\big)$$
  - ✓ $g_j : \mathbb{R}^{m_j} \rightarrow \bar{\mathbb{R}}$ are closed, proper, and convex.
  - ✓ $\mathbf{H}_j \in \mathbb{R}^{m_j \times n}$
- Can be re-written in canonical form

$$\min_{\mathbf{x}} \; f(\mathbf{x}) + g(\mathbf{H}\mathbf{x}),$$

with $\quad f = 0,$

Two or More Functions

- Problem template: $$\min_{\mathbf{x}\in\mathbb{R}^n} \sum_{j=1}^{J} g_j\big(\mathbf{H}_j\,\mathbf{x}\big)$$
  - ✓ $g_j : \mathbb{R}^{m_j} \to \bar{\mathbb{R}}$ are closed, proper, and convex.
  - ✓ $\mathbf{H}_j \in \mathbb{R}^{m_j\times n}$
- Can be re-written in canonical form

$$\min_{\mathbf{x}}\; f(\mathbf{x}) + g(\mathbf{H}\mathbf{x}),$$

with $f = 0,\; \mathbf{z} = \begin{bmatrix}\mathbf{z}^{(1)}\\ \vdots \\ \mathbf{z}^{(J)}\end{bmatrix},$

Two or More Functions

- Problem template: $$\min_{\mathbf{x}\in\mathbb{R}^n} \sum_{j=1}^{J} g_j\big(\mathbf{H}_j\,\mathbf{x}\big)$$
  - ✓ $g_j : \mathbb{R}^{m_j} \to \bar{\mathbb{R}}$ are closed, proper, and convex.
  - ✓ $\mathbf{H}_j \in \mathbb{R}^{m_j\times n}$
- Can be re-written in canonical form

$$\min_{\mathbf{x}} \; f(\mathbf{x}) + g(\mathbf{H}\mathbf{x}),$$

with $$f = 0, \;\; \mathbf{z} = \begin{bmatrix} \mathbf{z}^{(1)} \\ \vdots \\ \mathbf{z}^{(J)} \end{bmatrix}, \;\; g(\mathbf{z}) = \sum_{j=1}^{J} g_j(\mathbf{z}^{(j)}),$$

Two or More Functions

- Problem template: $$\min_{\mathbf{x}\in\mathbb{R}^n} \sum_{j=1}^{J} g_j\big(\mathbf{H}_j\,\mathbf{x}\big)$$

  ✓ $g_j : \mathbb{R}^{m_j} \rightarrow \bar{\mathbb{R}}$ are closed, proper, and convex.

  ✓ $\mathbf{H}_j \in \mathbb{R}^{m_j \times n}$

- Can be re-written in canonical form

$$\min_{\mathbf{x}}\; f(\mathbf{x}) + g(\mathbf{H}\mathbf{x}),$$

with $$f = 0,\; \mathbf{z} = \begin{bmatrix} \mathbf{z}^{(1)} \\ \vdots \\ \mathbf{z}^{(J)} \end{bmatrix},\; g(\mathbf{z}) = \sum_{j=1}^{J} g_j(\mathbf{z}^{(j)}),\; \mathbf{H} = \begin{bmatrix} \mathbf{H}_1 \\ \vdots \\ \mathbf{H}_J \end{bmatrix}$$

ADMM for Two or More Functions

- General problem template: $$\min_{\mathbf{x}\in\mathbb{R}^n} \sum_{j=1}^{J} g_j\big(\mathbf{H}_j\,\mathbf{x}\big)$$

ADMM for Two or More Functions

- General problem template: $$\min_{\mathbf{x}\in\mathbb{R}^n} \sum_{j=1}^{J} g_j\big(\mathbf{H}_j\,\mathbf{x}\big)$$

- ADMM after re-writing in canonical form:

$$\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}} \sum_{j=1}^{J} \|\mathbf{H}_j\,\mathbf{x} - \mathbf{z}_k^{(j)} + \mathbf{u}_k^{(j)}\|_2^2$$

ADMM for Two or More Functions

- General problem template: $$\min_{\mathbf{x}\in\mathbb{R}^n} \sum_{j=1}^{J} g_j\big(\mathbf{H}_j\,\mathbf{x}\big)$$

- ADMM after re-writing in canonical form:

$$\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}} \sum_{j=1}^{J} \|\mathbf{H}_j\,\mathbf{x} - \mathbf{z}_k^{(j)} + \mathbf{u}_k^{(j)}\|_2^2$$

$$\mathbf{z}_{k+1}^{(1)} = \arg\min_{\mathbf{v}\in\mathbb{R}^{m_1}} g_1(\mathbf{v}) + \frac{\rho}{2}\big\|\mathbf{H}_1\,\mathbf{x}_{k+1} - \mathbf{v} + \mathbf{u}_k^{(1)}\big\|_2^2$$

ADMM for Two or More Functions

- General problem template: $$\min_{\mathbf{x}\in\mathbb{R}^n} \sum_{j=1}^{J} g_j\big(\mathbf{H}_j\,\mathbf{x}\big)$$

- ADMM after re-writing in canonical form:

$$\begin{aligned}
\mathbf{x}_{k+1} &= \arg\min_{\mathbf{x}} \sum_{j=1}^{J} \|\mathbf{H}_j\,\mathbf{x} - \mathbf{z}_k^{(j)} + \mathbf{u}_k^{(j)}\|_2^2 \\
\mathbf{z}_{k+1}^{(1)} &= \arg\min_{\mathbf{v}\in\mathbb{R}^{m_1}} g_1(\mathbf{v}) + \frac{\rho}{2}\big\|\mathbf{H}_1\,\mathbf{x}_{k+1} - \mathbf{v} + \mathbf{u}_k^{(1)}\big\|_2^2 \\
\vdots \quad & \quad \vdots \\
\mathbf{z}_{k+1}^{(J)} &= \arg\min_{\mathbf{v}\in\mathbb{R}^{m_J}} g_J(\mathbf{v}) + \frac{\rho}{2}\big\|\mathbf{H}_J\,\mathbf{x}_{k+1} - \mathbf{v} + \mathbf{u}_k^{(J)}\big\|_2^2
\end{aligned}$$

ADMM for Two or More Functions

- General problem template: $$\min_{\mathbf{x}\in\mathbb{R}^n} \sum_{j=1}^{J} g_j\big(\mathbf{H}_j\,\mathbf{x}\big)$$
- ADMM after re-writing in canonical form:

$$\begin{aligned}
\mathbf{x}_{k+1} &= \arg\min_{\mathbf{x}} \sum_{j=1}^{J} \|\mathbf{H}_j\,\mathbf{x} - \mathbf{z}_k^{(j)} + \mathbf{u}_k^{(j)}\|_2^2 \\
\mathbf{z}_{k+1}^{(1)} &= \arg\min_{\mathbf{v}\in\mathbb{R}^{m_1}} g_1(\mathbf{v}) + \frac{\rho}{2}\big\|\mathbf{H}_1\,\mathbf{x}_{k+1} - \mathbf{v} + \mathbf{u}_k^{(1)}\big\|_2^2 \\
\vdots \quad & \quad \vdots \\
\mathbf{z}_{k+1}^{(J)} &= \arg\min_{\mathbf{v}\in\mathbb{R}^{m_J}} g_J(\mathbf{v}) + \frac{\rho}{2}\big\|\mathbf{H}_J\,\mathbf{x}_{k+1} - \mathbf{v} + \mathbf{u}_k^{(J)}\big\|_2^2 \\
\mathbf{u}_{k+1} &= \mathbf{u}_{k+1} + \mathbf{A}\mathbf{x}_{k+1} + \mathbf{B}\mathbf{z}_{k+1}
\end{aligned}$$

ADMM for Two or More Functions

- General problem template: $$\min_{\mathbf{x}\in\mathbb{R}^n} \sum_{j=1}^{J} g_j\big(\mathbf{H}_j\, \mathbf{x}\big)$$

- ADMM after re-writing in canonical form:

$$\begin{aligned}
\mathbf{x}_{k+1} &= \arg\min_{\mathbf{x}} \sum_{j=1}^{J} \|\mathbf{H}_j\,\mathbf{x} - \mathbf{z}_k^{(j)} + \mathbf{u}_k^{(j)}\|_2^2 \\
\mathbf{z}_{k+1}^{(1)} &= \arg\min_{\mathbf{v}\in\mathbb{R}^{m_1}} g_1(\mathbf{v}) + \frac{\rho}{2}\big\|\mathbf{H}_1\,\mathbf{x}_{k+1} - \mathbf{v} + \mathbf{u}_k^{(1)}\big\|_2^2 \\
&\vdots \qquad \vdots \\
\mathbf{z}_{k+1}^{(J)} &= \arg\min_{\mathbf{v}\in\mathbb{R}^{m_J}} g_J(\mathbf{v}) + \frac{\rho}{2}\big\|\mathbf{H}_J\,\mathbf{x}_{k+1} - \mathbf{v} + \mathbf{u}_k^{(J)}\big\|_2^2 \\
\mathbf{u}_{k+1} &= \mathbf{u}_{k+1} + \mathbf{A}\mathbf{x}_{k+1} + \mathbf{B}\mathbf{z}_{k+1}
\end{aligned}$$

- SALSA, PIDAL, PIDSplit, SDMM
  [Figueiredo and Bioucas-Dias, 2010], [Setzer et al., 2010], [Combettes and Pesquet, 2011]

A Closer Look

- A closer look at the algorithm

$$\mathbf{x}_{k+1} = \left(\sum_{j=1}^{J} \mathbf{H}_j^T \mathbf{H}_j\right)^{-1} \sum_{j=1}^{J} \mathbf{H}_j \left(\mathbf{z}_k^{(j)} - \mathbf{u}_k^{(j)}\right)$$

A Closer Look

- A closer look at the algorithm

$$
\begin{aligned}
\mathbf{x}_{k+1} &= \left(\sum_{j=1}^{J} \mathbf{H}_j^T \mathbf{H}_j\right)^{-1} \sum_{j=1}^{J} \mathbf{H}_j \big(\mathbf{z}_k^{(j)} - \mathbf{u}_k^{(j)}\big) \\
\mathbf{z}_{k+1}^{(1)} &= \mathrm{prox}_{g_1/\rho_k}\big(\mathbf{H}_1\, \mathbf{x}_{k+1} + \mathbf{u}_k^{(1)}\big) \\
\vdots \quad & \quad \vdots \\
\mathbf{z}_{k+1}^{(J)} &= \mathrm{prox}_{g_J/\rho_k}\big(\mathbf{H}_J\, \mathbf{x}_{k+1} + \mathbf{u}_k^{(J)}\big)
\end{aligned}
$$

A Closer Look

- A closer look at the algorithm

$$
\begin{aligned}
\mathbf{x}_{k+1} &= \left(\sum_{j=1}^{J} \mathbf{H}_j^T \mathbf{H}_j\right)^{-1} \sum_{j=1}^{J} \mathbf{H}_j \big(\mathbf{z}_k^{(j)} - \mathbf{u}_k^{(j)}\big)\\
\mathbf{z}_{k+1}^{(1)} &= \mathrm{prox}_{g_1/\rho_k}\big(\mathbf{H}_1\, \mathbf{x}_{k+1} + \mathbf{u}_k^{(1)}\big)\\
\vdots \quad & \quad \vdots\\
\mathbf{z}_{k+1}^{(J)} &= \mathrm{prox}_{g_J/\rho_k}\big(\mathbf{H}_J\, \mathbf{x}_{k+1} + \mathbf{u}_k^{(J)}\big)\\
\mathbf{u}_{k+1} &= \mathbf{u}_{k+1} + \mathbf{A}\mathbf{x}_{k+1} + \mathbf{B}\mathbf{z}_{k+1}
\end{aligned}
$$

A Closer Look

- A closer look at the algorithm

$$
\begin{aligned}
\mathbf{x}_{k+1} &= \left(\sum_{j=1}^{J} \mathbf{H}_j^T \mathbf{H}_j\right)^{-1} \sum_{j=1}^{J} \mathbf{H}_j \big(\mathbf{z}_k^{(j)} - \mathbf{u}_k^{(j)}\big) \\
\mathbf{z}_{k+1}^{(1)} &= \text{prox}_{g_1/\rho_k}\big(\mathbf{H}_1\, \mathbf{x}_{k+1} + \mathbf{u}_k^{(1)}\big) \\
\vdots \quad & \quad \vdots \\
\mathbf{z}_{k+1}^{(J)} &= \text{prox}_{g_J/\rho_k}\big(\mathbf{H}_J\, \mathbf{x}_{k+1} + \mathbf{u}_k^{(J)}\big) \\
\mathbf{u}_{k+1} &= \mathbf{u}_{k+1} + \mathbf{A}\mathbf{x}_{k+1} + \mathbf{B}\mathbf{z}_{k+1}
\end{aligned}
$$

- **Decoupled**: a linear problem; a set of proximity operators

A Closer Look

- A closer look at the algorithm

$$
\begin{aligned}
\mathbf{x}_{k+1} &= \left(\sum_{j=1}^{J} \mathbf{H}_j^T \mathbf{H}_j\right)^{-1} \sum_{j=1}^{J} \mathbf{H}_j \left(\mathbf{z}_k^{(j)} - \mathbf{u}_k^{(j)}\right) \\
\mathbf{z}_{k+1}^{(1)} &= \mathsf{prox}_{g_1/\rho_k}\left(\mathbf{H}_1\, \mathbf{x}_{k+1} + \mathbf{u}_k^{(1)}\right) \\
\vdots \quad & \quad \vdots \\
\mathbf{z}_{k+1}^{(J)} &= \mathsf{prox}_{g_J/\rho_k}\left(\mathbf{H}_J\, \mathbf{x}_{k+1} + \mathbf{u}_k^{(J)}\right) \\
\mathbf{u}_{k+1} &= \mathbf{u}_{k+1} + \mathbf{A}\mathbf{x}_{k+1} + \mathbf{B}\mathbf{z}_{k+1}
\end{aligned}
$$

- **Decoupled**: a linear problem; a set of proximity operators

- Hinges on: fast matrix inversion; simple proximity operators

A Closer Look

- A closer look at the algorithm

$$
\begin{aligned}
\mathbf{x}_{k+1} &= \left(\sum_{j=1}^{J} \mathbf{H}_j^T \mathbf{H}_j\right)^{-1} \sum_{j=1}^{J} \mathbf{H}_j \left(\mathbf{z}_k^{(j)} - \mathbf{u}_k^{(j)}\right) \\
\mathbf{z}_{k+1}^{(1)} &= \text{prox}_{g_1/\rho_k}\left(\mathbf{H}_1\, \mathbf{x}_{k+1} + \mathbf{u}_k^{(1)}\right) \\
\vdots \quad & \quad \vdots \\
\mathbf{z}_{k+1}^{(J)} &= \text{prox}_{g_J/\rho_k}\left(\mathbf{H}_J\, \mathbf{x}_{k+1} + \mathbf{u}_k^{(J)}\right) \\
\mathbf{u}_{k+1} &= \mathbf{u}_{k+1} + \mathbf{A}\mathbf{x}_{k+1} + \mathbf{B}\mathbf{z}_{k+1}
\end{aligned}
$$

- **Decoupled**: a linear problem; a set of proximity operators
- Hinges on: fast matrix inversion; simple proximity operators
- Matrix inverse independent of ρ_k (good, if not kept constant)

Outline

Image Restoration/Reconstruction

- General formulation: $$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x},\mathbf{y}) + \Phi(\mathbf{P}\mathbf{x}) + \iota_C(\mathbf{x})$$

where $\mathbf{y}$ are observations and $\iota_C(\mathbf{x}) = \begin{cases} 0 & \Leftarrow \quad \mathbf{x} \in C \\ +\infty & \Leftarrow \quad \mathbf{x} \notin C \end{cases}$

Image Restoration/Reconstruction

- General formulation: $$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x},\mathbf{y}) + \Phi(\mathbf{P}\mathbf{x}) + \iota_C(\mathbf{x})$$

  where $\mathbf{y}$ are observations and $\iota_C(\mathbf{x}) = \left\{\begin{array}{lcl} 0 & \Leftarrow & \mathbf{x}\in C \\ +\infty & \Leftarrow & \mathbf{x}\notin C \end{array}\right.$

- Ψ: the observation model (negative log-likelihood); namely,

Image Restoration/Reconstruction

- General formulation: $$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x},\mathbf{y}) + \Phi(\mathbf{P}\mathbf{x}) + \iota_C(\mathbf{x})$$

  where $\mathbf{y}$ are observations and $\iota_C(\mathbf{x}) = \begin{cases} 0 & \Leftarrow \quad \mathbf{x}\in C \\ +\infty & \Leftarrow \quad \mathbf{x}\notin C \end{cases}$

- Ψ: the observation model (negative log-likelihood); namely,
  - ✓ Gaussian observations: $\Psi(\mathbf{u},\mathbf{y}) = \frac{1}{2\sigma^2}\|\mathbf{u}-\mathbf{y}\|_2^2 = \frac{1}{2\sigma^2}\sum_i (u_i - y_i)^2$

Image Restoration/Reconstruction

- General formulation: $$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x},\mathbf{y}) + \Phi(\mathbf{P}\mathbf{x}) + \iota_C(\mathbf{x})$$

  where $\mathbf{y}$ are observations and $\iota_C(\mathbf{x}) = \begin{cases} 0 & \Leftarrow \quad \mathbf{x}\in C \\ +\infty & \Leftarrow \quad \mathbf{x}\notin C \end{cases}$

- Ψ: the observation model (negative log-likelihood); namely,
  - ✓ Gaussian observations: $\Psi(\mathbf{u},\mathbf{y}) = \frac{1}{2\sigma^2}\|\mathbf{u}-\mathbf{y}\|_2^2 = \frac{1}{2\sigma^2}\sum_i (u_i - y_i)^2$
  - ✓ Poisson observations: $\Psi(\mathbf{u},\mathbf{y}) = \sum_i \big(u_i + \iota_{\mathbb{R}_+}(u_i) - y_i \log(z_i)_+ \big)$

Image Restoration/Reconstruction

- General formulation: $$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x}, \mathbf{y}) + \Phi(\mathbf{P}\mathbf{x}) + \iota_C(\mathbf{x})$$

  where $\mathbf{y}$ are observations and $\iota_C(\mathbf{x}) = \begin{cases} 0 & \Leftarrow \quad \mathbf{x} \in C \\ +\infty & \Leftarrow \quad \mathbf{x} \notin C \end{cases}$

- Ψ: the observation model (negative log-likelihood); namely,
  - ✓ Gaussian observations: $\Psi(\mathbf{u}, \mathbf{y}) = \frac{1}{2\sigma^2}\|\mathbf{u} - \mathbf{y}\|_2^2 = \frac{1}{2\sigma^2}\sum_i (u_i - y_i)^2$
  - ✓ Poisson observations: $\Psi(\mathbf{u}, \mathbf{y}) = \sum_i \big(u_i + \iota_{\mathbb{R}_+}(u_i) - y_i \log(z_i)_+\big)$
  - ✓ Multiplicative noise: $\Psi(\mathbf{u}, \mathbf{y}) = M \sum_i \big(z_i + e^{y_i - z_i}\big)$

Image Restoration/Reconstruction

- General formulation: $$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x}, \mathbf{y}) + \Phi(\mathbf{P}\mathbf{x}) + \iota_C(\mathbf{x})$$

  where $\mathbf{y}$ are observations and $\iota_C(\mathbf{x}) = \begin{cases} 0 & \Leftarrow \quad \mathbf{x} \in C \\ +\infty & \Leftarrow \quad \mathbf{x} \notin C \end{cases}$

- Ψ: the observation model (negative log-likelihood); namely,
  - ✓ Gaussian observations: $\Psi(\mathbf{u}, \mathbf{y}) = \frac{1}{2\sigma^2}\|\mathbf{u} - \mathbf{y}\|_2^2 = \frac{1}{2\sigma^2}\sum_i (u_i - y_i)^2$
  - ✓ Poisson observations: $\Psi(\mathbf{u}, \mathbf{y}) = \sum_i \big(u_i + \iota_{\mathbb{R}_+}(u_i) - y_i \log(z_i)_+\big)$
  - ✓ Multiplicative noise: $\Psi(\mathbf{u}, \mathbf{y}) = M \sum_i \big(z_i + e^{y_i - z_i}\big)$

- $\Phi \circ \mathbf{P}$ is a regularizer; *e.g.*, total variation (TV), or Φ is a norm

Image Restoration/Reconstruction

- General formulation: $$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x},\mathbf{y}) + \Phi(\mathbf{P}\mathbf{x}) + \iota_C(\mathbf{x})$$

  where $\mathbf{y}$ are observations and $\iota_C(\mathbf{x}) = \begin{cases} 0 & \Leftarrow \quad \mathbf{x}\in C \\ +\infty & \Leftarrow \quad \mathbf{x}\notin C \end{cases}$

- Ψ: the observation model (negative log-likelihood); namely,
  - ✓ Gaussian observations: $\Psi(\mathbf{u},\mathbf{y}) = \frac{1}{2\sigma^2}\|\mathbf{u}-\mathbf{y}\|_2^2 = \frac{1}{2\sigma^2}\sum_i (u_i - y_i)^2$
  - ✓ Poisson observations: $\Psi(\mathbf{u},\mathbf{y}) = \sum_i \big(u_i + \iota_{\mathbb{R}_+}(u_i) - y_i \log(z_i)_+\big)$
  - ✓ Multiplicative noise: $\Psi(\mathbf{u},\mathbf{y}) = M\sum_i \big(z_i + e^{y_i - z_i}\big)$

- $\Phi \circ \mathbf{P}$ is a regularizer; *e.g.*, total variation (TV), or Φ is a norm

- $\mathbf{A}$: linear (observation) operator; *e.g.*, blur, tomographic projections, partial Fourier observations (MRI),...

Image Restoration: Observation Models Ψ

- General formulation: $$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x},\mathbf{y}) + \Phi(\mathbf{P}\mathbf{x}) + \iota_C(\mathbf{x})$$

Image Restoration: Observation Models Ψ

- General formulation: $$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x}, \mathbf{y}) + \Phi(\mathbf{P}\mathbf{x}) + \iota_C(\mathbf{x})$$

- All the above observation models have simple, component-wise proximity operators

Image Restoration: Observation Models Ψ

- General formulation: $$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x},\mathbf{y}) + \Phi(\mathbf{P}\mathbf{x}) + \iota_C(\mathbf{x})$$
- All the above observation models have simple, component-wise proximity operators
  - ✓ Gaussian observations: $\text{prox}_{\tau\Psi}(u) = \dfrac{\sigma^2 u + \tau y}{\sigma^2 + \tau}$

Image Restoration: Observation Models Ψ

- General formulation: $$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x}, \mathbf{y}) + \Phi(\mathbf{P}\mathbf{x}) + \iota_C(\mathbf{x})$$

- All the above observation models have simple, component-wise proximity operators
  - ✓ Gaussian observations: $\text{prox}_{\tau\Psi}(u) = \dfrac{\sigma^2 u + \tau y}{\sigma^2 + \tau}$
  - ✓ Poisson observations: $\text{prox}_{\tau\Psi}(u) = \dfrac{1}{2}\Big(y - \tau + \sqrt{(y-\tau)^2 + 4y\tau}\Big)$

Image Restoration: Observation Models Ψ

- General formulation: $$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x}, \mathbf{y}) + \Phi(\mathbf{P}\mathbf{x}) + \iota_C(\mathbf{x})$$
- All the above observation models have simple, component-wise proximity operators
  - ✓ Gaussian observations: $\text{prox}_{\tau\Psi}(u) = \dfrac{\sigma^2 u + \tau y}{\sigma^2 + \tau}$
  - ✓ Poisson observations: $\text{prox}_{\tau\Psi}(u) = \dfrac{1}{2}\Big(y - \tau + \sqrt{(y-\tau)^2 + 4y\tau}\Big)$
  - ✓ Multiplicative noise: $\text{prox}_{\tau\Psi}(u)$ uses Lambert's W-function

Image Restoration: Observation Models Ψ

- General formulation: $$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x},\mathbf{y}) + \Phi(\mathbf{P}\mathbf{x}) + \iota_C(\mathbf{x})$$

- All the above observation models have simple, component-wise proximity operators

  ✓ Gaussian observations: $\text{prox}_{\tau\Psi}(u) = \dfrac{\sigma^2 u + \tau y}{\sigma^2 + \tau}$

  ✓ Poisson observations: $\text{prox}_{\tau\Psi}(u) = \dfrac{1}{2}\Big(y - \tau + \sqrt{(y-\tau)^2 + 4y\tau}\Big)$

  ✓ Multiplicative noise: $\text{prox}_{\tau\Psi}(u)$ uses Lambert's W-function

- The proximity operator of ι_C is simply an Euclidean projection:

$$\text{prox}_{\iota_C}(\mathbf{u}) = \text{proj}_C(\mathbf{u});$$

Image Restoration: Observation Models Ψ

- General formulation: $$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x}, \mathbf{y}) + \Phi(\mathbf{P}\mathbf{x}) + \iota_C(\mathbf{x})$$

- All the above observation models have simple, component-wise proximity operators

  ✓ Gaussian observations: $\text{prox}_{\tau\Psi}(u) = \dfrac{\sigma^2 u + \tau y}{\sigma^2 + \tau}$

  ✓ Poisson observations: $\text{prox}_{\tau\Psi}(u) = \dfrac{1}{2}\Big(y - \tau + \sqrt{(y-\tau)^2 + 4y\tau}\Big)$

  ✓ Multiplicative noise: $\text{prox}_{\tau\Psi}(u)$ uses Lambert's W-function

- The proximity operator of ι_C is simply an Euclidean projection:

$$\text{prox}_{\iota_C}(\mathbf{u}) = \text{proj}_C(\mathbf{u});$$

e.g., if $C = \mathbb{R}^n_+$, then $\big(\text{proj}_C(\mathbf{u})\big)_i = \max\{0, u_i\}$

Image Restoration: Regularizers Φ

- General formulation: $$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x},\mathbf{y}) + \Phi(\mathbf{P}\mathbf{x}) + \iota_C(\mathbf{x})$$

Image Restoration: Regularizers Φ

- General formulation: $$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x}, \mathbf{y}) + \Phi(\mathbf{P}\mathbf{x}) + \iota_C(\mathbf{x})$$
- Classical regularizers with simple proximity operators

Image Restoration: Regularizers Φ

- General formulation: $\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x},\mathbf{y}) + \Phi(\mathbf{P}\mathbf{x}) + \iota_C(\mathbf{x})$

- Classical regularizers with simple proximity operators

  ✓ ℓ_1 norm: $\big(\mathsf{prox}_{\tau\|\cdot\|_1}(\mathbf{u})\big)_i = \mathsf{sign}(u_i)\max\{0, u_i - \tau\}$

Image Restoration: Regularizers Φ

- General formulation: $$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x},\mathbf{y}) + \Phi(\mathbf{P}\mathbf{x}) + \iota_C(\mathbf{x})$$

- Classical regularizers with simple proximity operators

  ✓ ℓ_1 norm: $\big(\text{prox}_{\tau\|\cdot\|_1}(\mathbf{u})\big)_i = \text{sign}(u_i)\max\{0, u_i - \tau\} = \text{soft}(u_i,\tau)$

Image Restoration: Regularizers Φ

- General formulation: $$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x},\mathbf{y}) + \Phi(\mathbf{P}\mathbf{x}) + \iota_C(\mathbf{x})$$

- Classical regularizers with simple proximity operators

  ✓ ℓ_1 norm: $\big(\mathsf{prox}_{\tau\|\cdot\|_1}(\mathbf{u})\big)_i = \mathsf{sign}(u_i)\max\{0, u_i - \tau\} = \mathsf{soft}(u_i,\tau)$

  ✓ Squared ℓ_2 norm: $\mathsf{prox}_{\tau\|\cdot\|_2^2}(\mathbf{u}) = \dfrac{\mathbf{u}}{1+\tau}$

Image Restoration: Regularizers Φ

- General formulation: $$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x},\mathbf{y}) + \Phi(\mathbf{P}\mathbf{x}) + \iota_C(\mathbf{x})$$

- Classical regularizers with simple proximity operators

  ✓ ℓ_1 norm: $\big(\mathsf{prox}_{\tau\|\cdot\|_1}(\mathbf{u})\big)_i = \mathsf{sign}(u_i)\max\{0, u_i - \tau\} = \mathsf{soft}(u_i,\tau)$

  ✓ Squared ℓ_2 norm: $\mathsf{prox}_{\tau\|\cdot\|_2^2}(\mathbf{u}) = \dfrac{\mathbf{u}}{1+\tau}$ (linear shrinkage)

Image Restoration: Regularizers Φ

- General formulation: $$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x}, \mathbf{y}) + \Phi(\mathbf{P}\mathbf{x}) + \iota_C(\mathbf{x})$$
- Classical regularizers with simple proximity operators
  - ✓ ℓ_1 norm: $\big(\text{prox}_{\tau\|\cdot\|_1}(\mathbf{u})\big)_i = \text{sign}(u_i)\max\{0, u_i - \tau\} = \text{soft}(u_i, \tau)$
  - ✓ Squared ℓ_2 norm: $\text{prox}_{\tau\|\cdot\|_2^2}(\mathbf{u}) = \dfrac{\mathbf{u}}{1+\tau}$ (linear shrinkage)
  - ✓ ℓ_2 norm: $\text{prox}_{\tau\|\cdot\|_2}(\mathbf{u}) = \dfrac{\mathbf{u}\ \max\{0, \|\mathbf{u}\|_2 - \tau\}}{\max\{0, \|\mathbf{u}\|_2 - \tau\} + \tau}$

Image Restoration: Regularizers Φ

- General formulation: $$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x},\mathbf{y}) + \Phi(\mathbf{P}\mathbf{x}) + \iota_C(\mathbf{x})$$

- Classical regularizers with simple proximity operators

  ✓ ℓ_1 norm: $\big(\text{prox}_{\tau\|\cdot\|_1}(\mathbf{u})\big)_i = \text{sign}(u_i)\max\{0, u_i - \tau\} = \text{soft}(u_i,\tau)$

  ✓ Squared ℓ_2 norm: $\text{prox}_{\tau\|\cdot\|_2^2}(\mathbf{u}) = \dfrac{\mathbf{u}}{1+\tau}$ (linear shrinkage)

  ✓ ℓ_2 norm: $\text{prox}_{\tau\|\cdot\|_2}(\mathbf{u}) = \dfrac{\mathbf{u}\ \max\{0, \|\mathbf{u}\|_2 - \tau\}}{\max\{0, \|\mathbf{u}\|_2 - \tau\} + \tau} = \text{vect-soft}(\mathbf{u},\tau)$

- Total variation can be written as $\Phi \circ \mathbf{P}$, where

$$\mathbf{P} : \mathbb{R}^n \rightarrow (\mathbb{R}^2)^n, \text{ with } (\mathbf{P}\mathbf{x})_i = \begin{bmatrix} x_i - x_{h(i)} \\ x_i - x_{v(i)} \end{bmatrix}, \text{ and } \Phi(\mathbf{v}) = \sum_i \|\mathbf{v}_i\|_2$$

with $h(i)$ and $v(i)$ the horizontal and vertical neighbours of pixel i

Image Restoration: Synthesis Formulation

- General formulation: $\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x}, \mathbf{y}) + \Phi(\mathbf{P}\mathbf{x})$

Image Restoration: Synthesis Formulation

- General formulation: $\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x}, \mathbf{y}) + \Phi(\mathbf{P}\mathbf{x})$

- Synthesis formulation: $\mathbf{A} = \mathbf{B}\mathbf{W}$ and $\mathbf{P} = \mathbf{I}$

Image Restoration: Synthesis Formulation

- General formulation: $\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x}, \mathbf{y}) + \Phi(\mathbf{P}\mathbf{x})$
- Synthesis formulation: $\mathbf{A} = \mathbf{B}\mathbf{W}$ and $\mathbf{P} = \mathbf{I}$
  - ✓ $\mathbf{x}$ contains representation coefficients, not the image itself

Image Restoration: Synthesis Formulation

- General formulation: $\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x}, \mathbf{y}) + \Phi(\mathbf{P}\mathbf{x})$
- Synthesis formulation: $\mathbf{A} = \mathbf{B}\mathbf{W}$ and $\mathbf{P} = \mathbf{I}$
  - ✓ $\mathbf{x}$ contains representation coefficients, not the image itself
  - ✓ $\mathbf{W}$ the synthesis operator of a Parseval frame: $\mathbf{W}\,\mathbf{W}^T = \mathbf{I}$

Image Restoration: Synthesis Formulation

- General formulation: $$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x}, \mathbf{y}) + \Phi(\mathbf{P}\mathbf{x})$$
- Synthesis formulation: $\mathbf{A} = \mathbf{B}\mathbf{W}$ and $\mathbf{P} = \mathbf{I}$
  - ✓ $\mathbf{x}$ contains representation coefficients, not the image itself
  - ✓ $\mathbf{W}$ the synthesis operator of a Parseval frame: $\mathbf{W}\,\mathbf{W}^T = \mathbf{I}$
  - ✓ $\mathbf{B}$ is the observation operator

Image Restoration: Synthesis Formulation

- General formulation: $\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x},\mathbf{y}) + \Phi(\mathbf{P}\mathbf{x})$
- Synthesis formulation: $\mathbf{A} = \mathbf{B}\mathbf{W}$ and $\mathbf{P} = \mathbf{I}$
  - ✓ $\mathbf{x}$ contains representation coefficients, not the image itself
  - ✓ $\mathbf{W}$ the synthesis operator of a Parseval frame: $\mathbf{W}\,\mathbf{W}^T = \mathbf{I}$
  - ✓ $\mathbf{B}$ is the observation operator
- Via the Sherman-Morrison-Woodbury formula

$$(\mathbf{A}^T\mathbf{A}+\mathbf{P}^T\mathbf{P})^{-1} = \left(\mathbf{W}^T\mathbf{B}^T\mathbf{B}\mathbf{W}+\mathbf{I}\right)^{-1} = \mathbf{I}-\mathbf{W}^T\mathbf{B}^T\left(\mathbf{B}^T\mathbf{B}+\mathbf{I}\right)^{-1}\mathbf{B}\mathbf{W}$$

Image Restoration: Synthesis Formulation

- General formulation: $\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x}, \mathbf{y}) + \Phi(\mathbf{P}\mathbf{x})$

- Synthesis formulation: $\mathbf{A} = \mathbf{B}\mathbf{W}$ and $\mathbf{P} = \mathbf{I}$
  - ✓ $\mathbf{x}$ contains representation coefficients, not the image itself
  - ✓ $\mathbf{W}$ the synthesis operator of a Parseval frame: $\mathbf{W}\,\mathbf{W}^T = \mathbf{I}$
  - ✓ $\mathbf{B}$ is the observation operator

- Via the Sherman-Morrison-Woodbury formula

$$(\mathbf{A}^T\mathbf{A}+\mathbf{P}^T\mathbf{P})^{-1} = \Big(\mathbf{W}^T\mathbf{B}^T\mathbf{B}\mathbf{W}+\mathbf{I}\Big)^{-1} = \mathbf{I}-\mathbf{W}^T\mathbf{B}^T\big(\mathbf{B}^T\mathbf{B}+\mathbf{I}\big)^{-1}\mathbf{B}\mathbf{W}$$

- Can $\mathbf{B}^T\mathbf{B}+\mathbf{I}$ be inverted efficiently?

Image Restoration: Analysis Formulation

- General formulation: $$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x}, \mathbf{y}) + \Phi(\mathbf{P}\mathbf{x})$$

Image Restoration: Analysis Formulation

- General formulation: $$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x}, \mathbf{y}) + \Phi(\mathbf{P}\mathbf{x})$$

- Analysis formulation: $\mathbf{A} = \mathbf{B}$

Image Restoration: Analysis Formulation

- General formulation: $\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x}, \mathbf{y}) + \Phi(\mathbf{P}\mathbf{x})$

- Analysis formulation: $\mathbf{A} = \mathbf{B}$

  ✓ $\mathbf{x}$ contains the image itself

Image Restoration: Analysis Formulation

- General formulation: $$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x}, \mathbf{y}) + \Phi(\mathbf{P}\mathbf{x})$$

- Analysis formulation: $\mathbf{A} = \mathbf{B}$

  ✓ $\mathbf{x}$ contains the image itself

  ✓ $\mathbf{P}$ the analysis operator of a Parseval frame: $\mathbf{P}^T\mathbf{P} = \mathbf{I}$

Image Restoration: Analysis Formulation

- General formulation: $$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x}, \mathbf{y}) + \Phi(\mathbf{P}\mathbf{x})$$

- Analysis formulation: $\mathbf{A} = \mathbf{B}$

  ✓ $\mathbf{x}$ contains the image itself

  ✓ $\mathbf{P}$ the analysis operator of a Parseval frame: $\mathbf{P}^T\mathbf{P} = \mathbf{I}$

  ✓ $\mathbf{B}$ is the observation operator

Image Restoration: Analysis Formulation

- General formulation: $\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x}, \mathbf{y}) + \Phi(\mathbf{P}\mathbf{x})$

- Analysis formulation: $\mathbf{A} = \mathbf{B}$

  ✓ $\mathbf{x}$ contains the image itself

  ✓ $\mathbf{P}$ the analysis operator of a Parseval frame: $\mathbf{P}^T\mathbf{P} = \mathbf{I}$

  ✓ $\mathbf{B}$ is the observation operator

- Matrix inversion:

$$\left(\mathbf{B}^T\mathbf{B} + \mathbf{P}^T\mathbf{P}\right)^{-1} = \left(\mathbf{B}^T\mathbf{B} + \mathbf{I}\right)^{-1}$$

Image Restoration: Analysis Formulation

- General formulation: $\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x}, \mathbf{y}) + \Phi(\mathbf{P}\mathbf{x})$

- Analysis formulation: $\mathbf{A} = \mathbf{B}$

  ✓ $\mathbf{x}$ contains the image itself

  ✓ $\mathbf{P}$ the analysis operator of a Parseval frame: $\mathbf{P}^T\mathbf{P} = \mathbf{I}$

  ✓ $\mathbf{B}$ is the observation operator

- Matrix inversion:

$$\left(\mathbf{B}^T\mathbf{B} + \mathbf{P}^T\mathbf{P}\right)^{-1} = \left(\mathbf{B}^T\mathbf{B} + \mathbf{I}\right)^{-1}$$

- Can $\mathbf{B}^T\mathbf{B} + \mathbf{I}$ be inverted efficiently?

Image Restoration: Constrained (Morozov) Formulations

- General formulation: $$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x}, \mathbf{y}) + \Phi(\mathbf{P}\mathbf{x})$$

Image Restoration: Constrained (Morozov) Formulations

- General formulation: $$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x}, \mathbf{y}) + \Phi(\mathbf{P}\mathbf{x})$$

- Constrained (or Morozov) formulation:

$$\hat{\mathbf{x}} \in \min_{\mathbf{x}\in\mathbb{R}^n} \Phi(\mathbf{P}\mathbf{x}) \;\text{ subject to }\; \Lambda(\mathbf{A}\mathbf{x}, \mathbf{y}) \leq 1$$

Image Restoration: Constrained (Morozov) Formulations

- General formulation: $\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x},\mathbf{y}) + \Phi(\mathbf{P}\mathbf{x})$

- Constrained (or Morozov) formulation:

$$\hat{\mathbf{x}} \in \min_{\mathbf{x}\in\mathbb{R}^n} \Phi(\mathbf{P}\mathbf{x}) \;\text{ subject to }\; \Lambda(\mathbf{A}\mathbf{x},\mathbf{y}) \leq 1$$

- Can be written in the general formulation, with

$$\Psi(\mathbf{A}\mathbf{x},\mathbf{y}) = \iota_{D(\mathbf{y})}(\mathbf{A}\mathbf{x}), \;\text{ with }\; D(\mathbf{y}) = \{\mathbf{x} : \Lambda(\mathbf{x},\mathbf{y}) \leq 1\}$$

Image Restoration: Constrained (Morozov) Formulations

- General formulation: $$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x},\mathbf{y}) + \Phi(\mathbf{P}\mathbf{x})$$

- Constrained (or Morozov) formulation:

$$\hat{\mathbf{x}} \in \min_{\mathbf{x}\in\mathbb{R}^n} \Phi(\mathbf{P}\mathbf{x}) \;\text{ subject to }\; \Lambda(\mathbf{A}\mathbf{x},\mathbf{y}) \leq 1$$

- Can be written in the general formulation, with

$$\Psi(\mathbf{A}\mathbf{x},\mathbf{y}) = \iota_{D(\mathbf{y})}(\mathbf{A}\mathbf{x}), \;\text{ with }\; D(\mathbf{y}) = \{\mathbf{x} : \Lambda(\mathbf{x},\mathbf{y}) \leq 1\}$$

- Classical example: $\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Phi(\mathbf{P}\mathbf{x})$ s.t. $\xi\|\mathbf{A}\mathbf{x}-\mathbf{y}\|_2 \leq 1$

  Thus, $D(\mathbf{y})$ is a unit Euclidean ball around $\mathbf{y}$; projection is trivial.

Image Restoration: Constrained (Morozov) Formulations

- General formulation: $\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{A}\mathbf{x},\mathbf{y}) + \Phi(\mathbf{P}\mathbf{x})$

- Constrained (or Morozov) formulation:

$$\hat{\mathbf{x}} \in \min_{\mathbf{x}\in\mathbb{R}^n} \Phi(\mathbf{P}\mathbf{x}) \text{ subject to } \Lambda(\mathbf{A}\mathbf{x},\mathbf{y}) \leq 1$$

- Can be written in the general formulation, with

$$\Psi(\mathbf{A}\mathbf{x},\mathbf{y}) = \iota_{D(\mathbf{y})}(\mathbf{A}\mathbf{x}), \text{ with } D(\mathbf{y}) = \{\mathbf{x} : \Lambda(\mathbf{x},\mathbf{y}) \leq 1\}$$

- Classical example: $\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Phi(\mathbf{P}\mathbf{x})$ s.t. $\xi\|\mathbf{A}\mathbf{x}-\mathbf{y}\|_2 \leq 1$

  Thus, $D(\mathbf{y})$ is a unit Euclidean ball around $\mathbf{y}$; projection is trivial.

- Applies both to synthesis and analysis formulations

Image Restoration: Matrix Inversions

- The required inversion $\left(\mathbf{B}^T\mathbf{B} + \mathbf{I}\right)^{-1}$ is simple in many relevant cases: [Afonso et al., 2011], [Figueiredo and Bioucas-Dias, 2010]

Image Restoration: Matrix Inversions

- The required inversion $\left(\mathbf{B}^T\mathbf{B}+\mathbf{I}\right)^{-1}$ is simple in many relevant cases: [Afonso et al., 2011], [Figueiredo and Bioucas-Dias, 2010]

  ✓ Periodic deconvolution: $\mathbf{B} = \mathbf{U}^H\mathbf{F}\mathbf{U}$,

  $\mathbf{F}$ is diagonal; $\mathbf{U}$ is the DFT matrix ($\mathbf{U}^H\mathbf{U} = \mathbf{U}\mathbf{U}^H = \mathbf{I}$)

$$\left(\mathbf{B}^T\mathbf{B}+\mathbf{I}\right)^{-1} = \mathbf{U}^H \underbrace{\left(|\mathbf{F}|^2+\mathbf{I}\right)^{-1}}_{\text{diagonal}} \mathbf{U}$$

Image Restoration: Matrix Inversions

- The required inversion $\left(\mathbf{B}^T\mathbf{B} + \mathbf{I}\right)^{-1}$ is simple in many relevant cases: [Afonso et al., 2011], [Figueiredo and Bioucas-Dias, 2010]

  ✓ Periodic deconvolution: $\mathbf{B} = \mathbf{U}^H \mathbf{F} \mathbf{U}$,

  $\mathbf{F}$ is diagonal; $\mathbf{U}$ is the DFT matrix ($\mathbf{U}^H\mathbf{U} = \mathbf{U}\mathbf{U}^H = \mathbf{I}$)

$$\left(\mathbf{B}^T\mathbf{B} + \mathbf{I}\right)^{-1} = \mathbf{U}^H \underbrace{\left(|\mathbf{F}|^2 + \mathbf{I}\right)^{-1}}_{\text{diagonal}} \mathbf{U}$$

  ✓ Inpainting: $\mathbf{B} \in \{0,1\}^{m\times n}$, with m rows of $\mathbf{I}$; thus, $\mathbf{B}^T\mathbf{B}$ is diagonal

Image Restoration: Matrix Inversions

- The required inversion $\left(\mathbf{B}^T\mathbf{B}+\mathbf{I}\right)^{-1}$ is simple in many relevant cases: [Afonso et al., 2011], [Figueiredo and Bioucas-Dias, 2010]

  ✓ Periodic deconvolution: $\mathbf{B} = \mathbf{U}^H\mathbf{F}\mathbf{U}$,

  $\mathbf{F}$ is diagonal; $\mathbf{U}$ is the DFT matrix $(\mathbf{U}^H\mathbf{U} = \mathbf{U}\mathbf{U}^H = \mathbf{I})$

$$\left(\mathbf{B}^T\mathbf{B}+\mathbf{I}\right)^{-1} = \mathbf{U}^H \underbrace{\left(|\mathbf{F}|^2+\mathbf{I}\right)^{-1}}_{\text{diagonal}} \mathbf{U}$$

  ✓ Inpainting: $\mathbf{B} \in \{0,1\}^{m\times n}$, with m rows of $\mathbf{I}$; thus, $\mathbf{B}^T\mathbf{B}$ is diagonal

  ✓ Compressive Fourier imaging (MRI, multi-coil MRI): $\mathbf{B} = \mathbf{M}\mathbf{U}$, where $\mathbf{M} \in \{0,1\}^{m\times n}$, with m rows of $\mathbf{I}$; thus, $\mathbf{M}\mathbf{M}^T = \mathbf{I}$

$$\left(\mathbf{B}^T\mathbf{B}+\mathbf{I}\right)^{-1} = \mathbf{I} - \frac{1}{2}\,\mathbf{U}^H \underbrace{\mathbf{M}^T\mathbf{M}}_{\text{diagonal}} \mathbf{U}$$

Image Restoration: Matrix Inversions

- The required inversion $\left(\mathbf{B}^T\mathbf{B}+\mathbf{I}\right)^{-1}$ is simple in many relevant cases: [Afonso et al., 2011], [Figueiredo and Bioucas-Dias, 2010]

  ✓ Periodic deconvolution: $\mathbf{B} = \mathbf{U}^H\mathbf{F}\mathbf{U}$,

  $\mathbf{F}$ is diagonal; $\mathbf{U}$ is the DFT matrix ($\mathbf{U}^H\mathbf{U} = \mathbf{U}\mathbf{U}^H = \mathbf{I}$)

$$\left(\mathbf{B}^T\mathbf{B}+\mathbf{I}\right)^{-1} = \mathbf{U}^H \underbrace{\left(|\mathbf{F}|^2+\mathbf{I}\right)^{-1}}_{\text{diagonal}} \mathbf{U}$$

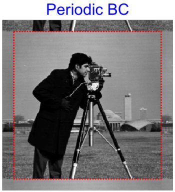
  ✓ Inpainting: $\mathbf{B} \in \{0,1\}^{m\times n}$, with m rows of $\mathbf{I}$; thus, $\mathbf{B}^T\mathbf{B}$ is diagonal

  ✓ Compressive Fourier imaging (MRI, multi-coil MRI): $\mathbf{B} = \mathbf{M}\mathbf{U}$, where $\mathbf{M} \in \{0,1\}^{m\times n}$, with m rows of $\mathbf{I}$; thus, $\mathbf{M}\mathbf{M}^T = \mathbf{I}$

$$\left(\mathbf{B}^T\mathbf{B}+\mathbf{I}\right)^{-1} = \mathbf{I} - \frac{1}{2}\,\mathbf{U}^H \underbrace{\mathbf{M}^T\mathbf{M}}_{\text{diagonal}}\ \mathbf{U}$$

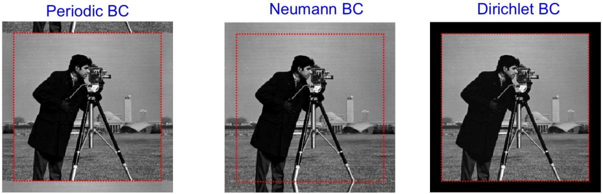
- Cost is at most $O(n \log n)$

Non-periodic Deconvolution



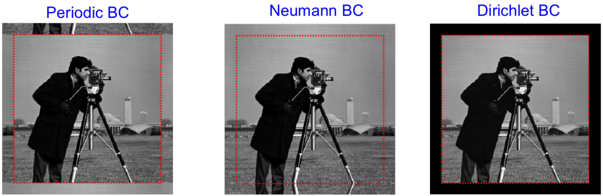
- Periodic boundary conditions are usually unnatural

Non-periodic Deconvolution

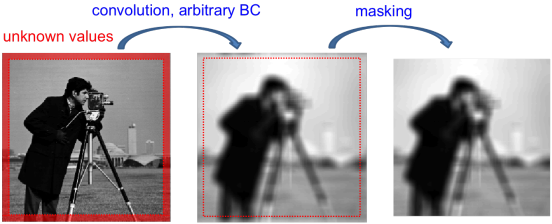


- Periodic boundary conditions are usually unnatural
- ...as are other standard BC: Neumann, Dirichlet.

Non-periodic Deconvolution



- Periodic boundary conditions are usually unnatural
- ...as are other standard BC: Neumann, Dirichlet.
- A more natural choice: unknown boundaries [Reeves, 2005], [Chan et al., 2005], [Almeida and Figueiredo, 2013a], [Ramani and Fessler, 2013]



Non-periodic Deconvolution (2)

- Gaussian noise model: $\Psi(\mathbf{B}\mathbf{x}, \mathbf{y}) = \frac{1}{2\sigma^2} \| \overbrace{\mathbf{M}}^{\text{mask}} \underbrace{\mathbf{U}^H \mathbf{F} \mathbf{U}}_{\text{period. conv.}} \mathbf{x} - \mathbf{y} \|_2^2$

Non-periodic Deconvolution (2)

- Gaussian noise model: $\Psi(\mathbf{Bx}, \mathbf{y}) = \frac{1}{2\sigma^2} \| \overbrace{\mathbf{M}}^{\text{mask}} \underbrace{\mathbf{U}^H \mathbf{F} \mathbf{U}}_{\text{period. conv.}} \mathbf{x} - \mathbf{y} \|_2^2$
- Choosing $\mathbf{B} = \mathbf{M}\mathbf{U}^H\mathbf{F}\mathbf{U}$, makes $\left(\mathbf{B}^T\mathbf{B} + \mathbf{I}\right)^{-1}$ hard to compute

Non-periodic Deconvolution (2)

- Gaussian noise model: $\Psi(\mathbf{Bx},\mathbf{y}) = \frac{1}{2\sigma^2}\|\overbrace{\mathbf{M}}^{\text{mask}}\underbrace{\mathbf{U}^H\mathbf{F}\mathbf{U}}_{\text{period. conv.}}\mathbf{x}-\mathbf{y}\|_2^2$

- Choosing $\mathbf{B} = \mathbf{M}\mathbf{U}^H\mathbf{F}\mathbf{U}$, makes $\left(\mathbf{B}^T\mathbf{B}+\mathbf{I}\right)^{-1}$ hard to compute

- Better option: $\mathbf{B} = \mathbf{U}^H\mathbf{F}\mathbf{U}$ (as in periodic deconvolution), and

$$\Psi(\mathbf{u},\mathbf{y}) = \frac{1}{2\sigma^2}\|\mathbf{Mu}-\mathbf{y}\|_2^2$$

Non-periodic Deconvolution (2)

- Gaussian noise model: $\Psi(\mathbf{Bx}, \mathbf{y}) = \frac{1}{2\sigma^2}\| \overbrace{\mathbf{M}}^{\text{mask}} \underbrace{\mathbf{U}^H\mathbf{F}\mathbf{U}}_{\text{period. conv.}} \mathbf{x} - \mathbf{y}\|_2^2$
- Choosing $\mathbf{B} = \mathbf{M}\mathbf{U}^H\mathbf{F}\mathbf{U}$, makes $\left(\mathbf{B}^T\mathbf{B} + \mathbf{I}\right)^{-1}$ hard to compute
- Better option: $\mathbf{B} = \mathbf{U}^H\mathbf{F}\mathbf{U}$ (as in periodic deconvolution), and

$$\Psi(\mathbf{u}, \mathbf{y}) = \frac{1}{2\sigma^2}\|\mathbf{M}\mathbf{u} - \mathbf{y}\|_2^2$$

- The proximity operator is still simple:

$$\text{prox}_{\tau\Psi}(\mathbf{u}) = \underbrace{\left(\tau\mathbf{M}^T\mathbf{M} + \sigma^2\mathbf{I}\right)^{-1}}_{\text{diagonal}}\left(\tau\mathbf{M}^T\mathbf{y} + \sigma^2\mathbf{u}\right)$$

Non-periodic Deconvolution (2)

- Gaussian noise model: $\Psi(\mathbf{Bx}, \mathbf{y}) = \frac{1}{2\sigma^2}\| \overbrace{\mathbf{M}}^{\text{mask}} \underbrace{\mathbf{U}^H\mathbf{F}\mathbf{U}}_{\text{period. conv.}} \mathbf{x} - \mathbf{y}\|_2^2$

- Choosing $\mathbf{B} = \mathbf{M}\mathbf{U}^H\mathbf{F}\mathbf{U}$, makes $\left(\mathbf{B}^T\mathbf{B} + \mathbf{I}\right)^{-1}$ hard to compute

- Better option: $\mathbf{B} = \mathbf{U}^H\mathbf{F}\mathbf{U}$ (as in periodic deconvolution), and

$$\Psi(\mathbf{u}, \mathbf{y}) = \frac{1}{2\sigma^2}\|\mathbf{Mu} - \mathbf{y}\|_2^2$$

- The proximity operator is still simple:

$$\text{prox}_{\tau\Psi}(\mathbf{u}) = \underbrace{\left(\tau\mathbf{M}^T\mathbf{M} + \sigma^2\mathbf{I}\right)^{-1}}_{\text{diagonal}}\left(\tau\mathbf{M}^T\mathbf{y} + \sigma^2\mathbf{u}\right)$$

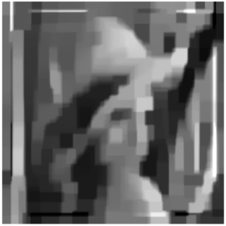
- Similar formulations:
  - ✓ deconvolution + inpainting ($\mathbf{M}$ masks the boundary and missing pixels)
  - ✓ super-resolution (filtering + downsampling mask)

Deconvolution with Unknown Boundaries: Example

original (256 × 256)

observed (238 × 238)

Assuming periodic BC



FA-BC (ISNR = -2.52dB)

Edge tapering

FA-ET (ISNR = 3.06dB)

Unknown BC by ADMM

FA-MD (ISRN = 10.63dB)

Deconvolution + Inpainting with Unknown BC: Example

original (256×256)

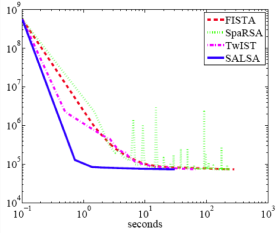
observed (238×238)

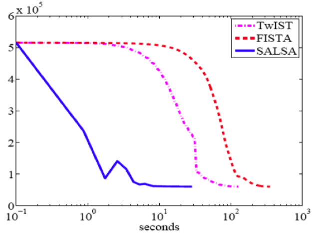
FA-CG (SNR = 20.58dB)

FA-MD (SNR = 20.57dB)

Speed

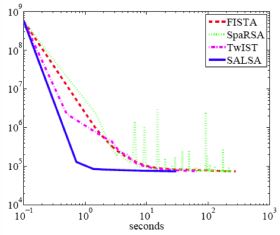
- Benchmark deblurring problem (9×9 blur, 40dB SNR, Haar frame, ℓ_1) and inpainting problem (50% missing data) [Afonso et al., 2011]

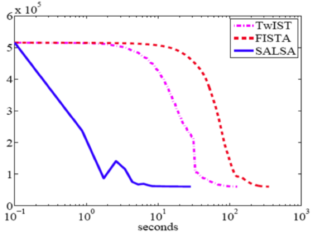




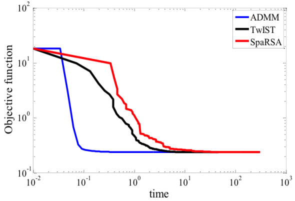
Speed

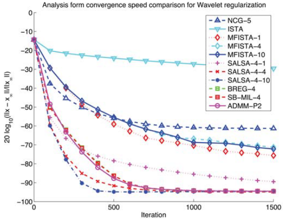
- Benchmark deblurring problem (9×9 blur, 40dB SNR, Haar frame, ℓ_1) and inpainting problem (50% missing data) [Afonso et al., 2011]





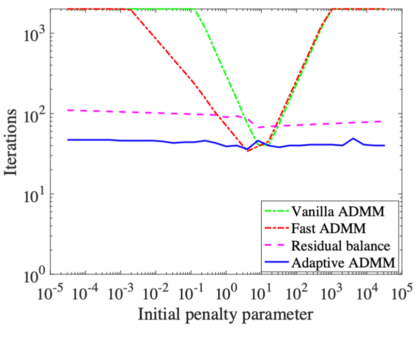
- Deconvolution with unknown BC [Almeida and Figueiredo, 2013a], [Ramani and Fessler, 2013]





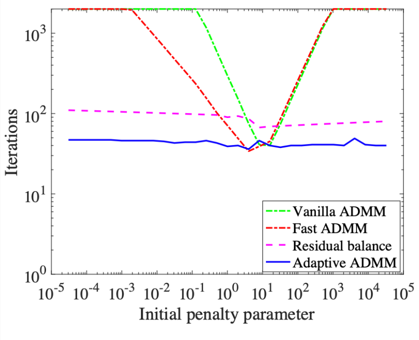
Speed

- Key issue: choosing parameter ρ_k
- Barzilai-Borwein-type method on the dual [Xu et al., 2016]



Speed

- Key issue: choosing parameter ρ_k
- Barzilai-Borwein-type method on the dual [Xu et al., 2016]



- Extension to over-relaxed and distributed ADMM [Xu et al., 2017a,b]

Intermediate Summary

- ...a flexible formulation of ADMM for image restoration:

Intermediate Summary

- ...a flexible formulation of ADMM for image restoration:
  - ✓ Frame-based analysis or synthesis regularization

Intermediate Summary

- ...a flexible formulation of ADMM for image restoration:
  - ✓ Frame-based analysis or synthesis regularization
  - ✓ Total variation regularization

Intermediate Summary

- ...a flexible formulation of ADMM for image restoration:
  - ✓ Frame-based analysis or synthesis regularization
  - ✓ Total variation regularization
  - ✓ ...or combinations thereof

Intermediate Summary

- ...a flexible formulation of ADMM for image restoration:
  - ✓ Frame-based analysis or synthesis regularization
  - ✓ Total variation regularization
  - ✓ ...or combinations thereof
  - ✓ Tikhonov, Morozov, Ivanov formulations

Intermediate Summary

- ...a flexible formulation of ADMM for image restoration:
  - ✓ Frame-based analysis or synthesis regularization
  - ✓ Total variation regularization
  - ✓ ...or combinations thereof
  - ✓ Tikhonov, Morozov, Ivanov formulations
  - ✓ Gaussian, Poissonian, multiplicative noise, ...

Intermediate Summary

- ...a flexible formulation of ADMM for image restoration:
  - ✓ Frame-based analysis or synthesis regularization
  - ✓ Total variation regularization
  - ✓ ...or combinations thereof
  - ✓ Tikhonov, Morozov, Ivanov formulations
  - ✓ Gaussian, Poissonian, multiplicative noise, ...
  - ✓ Deconvolution, inpainting, compressive Fourier sensing (MRI), super-resolution, ...

Intermediate Summary

- ...a flexible formulation of ADMM for image restoration:
  - ✓ Frame-based analysis or synthesis regularization
  - ✓ Total variation regularization
  - ✓ ...or combinations thereof
  - ✓ Tikhonov, Morozov, Ivanov formulations
  - ✓ Gaussian, Poissonian, multiplicative noise, ...
  - ✓ Deconvolution, inpainting, compressive Fourier sensing (MRI), super-resolution, ...
  - ✓ Periodic or unknown boundaries

Intermediate Summary

- ...a flexible formulation of ADMM for image restoration:
  - ✓ Frame-based analysis or synthesis regularization
  - ✓ Total variation regularization
  - ✓ ...or combinations thereof
  - ✓ Tikhonov, Morozov, Ivanov formulations
  - ✓ Gaussian, Poissonian, multiplicative noise, ...
  - ✓ Deconvolution, inpainting, compressive Fourier sensing (MRI), super-resolution, ...
  - ✓ Periodic or unknown boundaries
  - ✓ Blind deconvolution (later)
- Convergence guaranteed by classical results [Eckstein and Bertsekas, 1992] ...functions are closed, proper, convex; matrices have full column rank (except blind deconvolution)

Outline

1. Introduction: ADMM et al. (2007-2011)
2. Image Restoration/Reconstruction (2011-2014)
3. Plug-and-Play and Class-Adaptation (2015-2020)
4. Blind Restoration: Non-Convex Optimization (2013-2019)
5. Hyperspectral Imaging (2017-2020)
6. Final Remarks

Denoising Step in ADMM

- Restoration (w/ Gauss noise): $\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \frac{1}{2}\|\mathbf{A}\mathbf{x}-\mathbf{y}\|_2^2 + \Phi(\mathbf{x})$

Denoising Step in ADMM

- Restoration (w/ Gauss noise): $\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \Phi(\mathbf{x})$
- ADMM directly applied to this problem has the form

$$
\begin{aligned}
\mathbf{x}_{k+1} &= \big(\mathbf{A}^T\mathbf{A} + \rho\mathbf{I}\big)^{-1}\big(\mathbf{A}^T\mathbf{y} + \rho(\mathbf{z}_k - \mathbf{u}_k)\big)\\
\mathbf{z}_{k+1} &= \mathsf{prox}_{\Phi/\rho}\big(\mathbf{x}_{k+1} + \mathbf{u}_k\big)\\
\mathbf{u}_{k+1} &= \mathbf{u}_{k+1} + \mathbf{x}_{k+1} - \mathbf{z}_{k+1}
\end{aligned}
$$

Denoising Step in ADMM

- Restoration (w/ Gauss noise): $\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \Phi(\mathbf{x})$
- ADMM directly applied to this problem has the form

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

$$\mathbf{z}_{k+1} = \mathsf{prox}_{\Phi/\rho}\big(\mathbf{x}_{k+1} + \mathbf{u}_k\big)$$

$$\mathbf{u}_{k+1} = \mathbf{u}_{k+1} + \mathbf{x}_{k+1} - \mathbf{z}_{k+1}$$

- The prox of the regularizer Φ is a denoising operation

Denoising Step in ADMM

- Restoration (w/ Gauss noise): $\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \Phi(\mathbf{x})$

- ADMM directly applied to this problem has the form

$$\mathbf{x}_{k+1} = \left(\mathbf{A}^T\mathbf{A} + \rho\mathbf{I}\right)^{-1}\left(\mathbf{A}^T\mathbf{y} + \rho(\mathbf{z}_k - \mathbf{u}_k)\right)$$
$$\mathbf{z}_{k+1} = \mathsf{prox}_{\Phi/\rho}\left(\mathbf{x}_{k+1} + \mathbf{u}_k\right)$$
$$\mathbf{u}_{k+1} = \mathbf{u}_{k+1} + \mathbf{x}_{k+1} - \mathbf{z}_{k+1}$$

- The prox of the regularizer Φ is a denoising operation

- Prox of convex regularizer (frames, TV): not state-of-the-art denoising

Denoising Step in ADMM

- Restoration (w/ Gauss noise): $\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \Phi(\mathbf{x})$
- ADMM directly applied to this problem has the form

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

- The prox of the regularizer Φ is a denoising operation
- Prox of convex regularizer (frames, TV): not state-of-the-art denoising
- State-of-the-art denoising methods:

Denoising Step in ADMM

- Restoration (w/ Gauss noise): $\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \frac{1}{2}\|\mathbf{A}\mathbf{x}-\mathbf{y}\|_2^2 + \Phi(\mathbf{x})$
- ADMM directly applied to this problem has the form

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

- The prox of the regularizer Φ is a denoising operation
- Prox of convex regularizer (frames, TV): not state-of-the-art denoising
- State-of-the-art denoising methods:
  - ✓ Collaborative filtering (BM3D) [Dabov et al., 2007]
  - ✓ Non-local Bayes [Lebrun et al., 2013]
  - ✓ Gaussian mixture models [Zoran and Weiss, 2011], [Teodoro et al., 2015]
  - ✓ Deep convolutional networks [Ulyanov et al., 2017, Heckel and Hand, 2019]

Denoising Step in ADMM

- Restoration (w/ Gauss noise): $\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \Phi(\mathbf{x})$
- ADMM directly applied to this problem has the form

$$\mathbf{x}_{k+1} = \left(\mathbf{A}^T\mathbf{A} + \rho\mathbf{I}\right)^{-1}\left(\mathbf{A}^T\mathbf{y} + \rho(\mathbf{z}_k - \mathbf{u}_k)\right)$$
$$\mathbf{z}_{k+1} = \mathsf{prox}_{\Phi/\rho}\left(\mathbf{x}_{k+1} + \mathbf{u}_k\right)$$
$$\mathbf{u}_{k+1} = \mathbf{u}_{k+1} + \mathbf{x}_{k+1} - \mathbf{z}_{k+1}$$

- The prox of the regularizer Φ is a denoising operation
- Prox of convex regularizer (frames, TV): not state-of-the-art denoising
- State-of-the-art denoising methods:
  - ✓ Collaborative filtering (BM3D) [Dabov et al., 2007]
  - ✓ Non-local Bayes [Lebrun et al., 2013]
  - ✓ Gaussian mixture models [Zoran and Weiss, 2011], [Teodoro et al., 2015]
  - ✓ Deep convolutional networks [Ulyanov et al., 2017, Heckel and Hand, 2019]
- Can we use one of these denoisers instead of a proximity operator?

Plug-and-Play ADMM

- Plug a black-box denoiser into ADMM [Venkatakrishnan et al., 2013]

$$\mathbf{x}_{k+1} = \left(\mathbf{A}^T\mathbf{A} + \rho\mathbf{I}\right)^{-1}\left(\mathbf{A}^T\mathbf{y} + \rho(\mathbf{z}_k - \mathbf{u}_k)\right)$$
$$\mathbf{z}_{k+1} = \text{denoise}\left(\mathbf{x}_{k+1} + \mathbf{u}_k\right)$$
$$\mathbf{u}_{k+1} = \mathbf{u}_k + \mathbf{x}_{k+1} - \mathbf{z}_{k+1}$$

Plug-and-Play ADMM

- Plug a black-box denoiser into ADMM [Venkatakrishnan et al., 2013]

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

- If denoiser $= \text{prox}_\phi$, for convex ϕ, convergence is well-known [Eckstein and Bertsekas, 1992, Boyd et al., 2011, ..., ...].

Plug-and-Play ADMM

- Plug a black-box denoiser into ADMM [Venkatakrishnan et al., 2013]

$$\mathbf{x}_{k+1} = \left(\mathbf{A}^T\mathbf{A} + \rho\mathbf{I}\right)^{-1}\left(\mathbf{A}^T\mathbf{y} + \rho(\mathbf{z}_k - \mathbf{u}_k)\right)$$
$$\mathbf{z}_{k+1} = \text{denoise}\left(\mathbf{x}_{k+1} + \mathbf{u}_k\right)$$
$$\mathbf{u}_{k+1} = \mathbf{u}_k + \mathbf{x}_{k+1} - \mathbf{z}_{k+1}$$

- If denoiser $= \text{prox}_\phi$, for convex ϕ, convergence is well-known [Eckstein and Bertsekas, 1992, Boyd et al., 2011, ..., ...].

- ...what about convergence of PnP-ADMM? [Sreehari et al., 2016, Teodoro et al., 2017b, 2019, Chan et al., 2017, Xu et al., 2020, ..., ...]

GMM-Based Denoising

- Observation model: $p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{x}, \sigma^2\mathbf{I})$

GMM-Based Denoising

- Observation model: $p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{x}, \sigma^2\mathbf{I})$
- Decompose noisy image into overlapping patches $\mathbf{y}_i$

GMM-Based Denoising

- Observation model: $p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{x}, \sigma^2\mathbf{I})$
- Decompose noisy image into overlapping patches $\mathbf{y}_i$
- Denoise each patch independently under GMM prior:

$$p(\mathbf{x}_i) = \sum_{j=1}^{K} \alpha_j \, \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_j, \mathbf{C}_j)$$

GMM-Based Denoising

- Observation model: $p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{x}, \sigma^2\mathbf{I})$
- Decompose noisy image into overlapping patches $\mathbf{y}_i$
- Denoise each patch independently under GMM prior:

$$p(\mathbf{x}_i) = \sum_{j=1}^{K} \alpha_j \, \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_j, \mathbf{C}_j)$$

- The minimum mean squared error (MMSE) estimate (not the MAP) has closed-form:

$$\hat{\mathbf{x}}_i = \mathbb{E}[\mathbf{X}_i|\mathbf{y}_i]$$

GMM-Based Denoising

- Observation model: $p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{x}, \sigma^2\mathbf{I})$
- Decompose noisy image into overlapping patches $\mathbf{y}_i$
- Denoise each patch independently under GMM prior:

$$p(\mathbf{x}_i) = \sum_{j=1}^{K} \alpha_j \, \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_j, \mathbf{C}_j)$$

- The minimum mean squared error (MMSE) estimate (not the MAP) has closed-form:

$$\hat{\mathbf{x}}_i = \mathbb{E}[\mathbf{X}_i|\mathbf{y}_i]$$

- Assemble the denoised image by putting the estimated patches at their locations, averaging overlapping pixel estimates

GMM-Based Denoising

- Observation model: $p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{x}, \sigma^2\mathbf{I})$
- Decompose noisy image into overlapping patches $\mathbf{y}_i$
- Denoise each patch independently under GMM prior:

$$p(\mathbf{x}_i) = \sum_{j=1}^{K} \alpha_j \, \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_j, \mathbf{C}_j)$$

- The minimum mean squared error (MMSE) estimate (not the MAP) has closed-form:

$$\hat{\mathbf{x}}_i = \mathbb{E}[\mathbf{X}_i|\mathbf{y}_i]$$

- Assemble the denoised image by putting the estimated patches at their locations, averaging overlapping pixel estimates
- Estimating the mixture:

GMM-Based Denoising

- Observation model: $p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{x}, \sigma^2\mathbf{I})$
- Decompose noisy image into overlapping patches $\mathbf{y}_i$
- Denoise each patch independently under GMM prior:

$$p(\mathbf{x}_i) = \sum_{j=1}^{K} \alpha_j \, \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_j, \mathbf{C}_j)$$

- The minimum mean squared error (MMSE) estimate (not the MAP) has closed-form:

$$\hat{\mathbf{x}}_i = \mathbb{E}[\mathbf{X}_i|\mathbf{y}_i]$$

- Assemble the denoised image by putting the estimated patches at their locations, averaging overlapping pixel estimates
- Estimating the mixture:
  - ✓ From a collection of clean image patches [Zoran and Weiss, 2011]

GMM-Based Denoising

- Observation model: $p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{x}, \sigma^2\mathbf{I})$
- Decompose noisy image into overlapping patches $\mathbf{y}_i$
- Denoise each patch independently under GMM prior:

$$p(\mathbf{x}_i) = \sum_{j=1}^{K} \alpha_j \, \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_j, \mathbf{C}_j)$$

- The minimum mean squared error (MMSE) estimate (not the MAP) has closed-form:

$$\hat{\mathbf{x}}_i = \mathbb{E}[\mathbf{X}_i|\mathbf{y}_i]$$

- Assemble the denoised image by putting the estimated patches at their locations, averaging overlapping pixel estimates
- Estimating the mixture:
  - ✓ From a collection of clean image patches [Zoran and Weiss, 2011]
  - ✓ From the noisy image itself using EM [Teodoro et al., 2015]

MMSE Estimate with GMM Prior

- Gaussian noisy observations: $f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{x}, \sigma^2\mathbf{I})$

MMSE Estimate with GMM Prior

- Gaussian noisy observations: $f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{x}, \sigma^2\mathbf{I})$
- Gaussian prior: $f_{\mathbf{X}}(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{C})$

MMSE Estimate with GMM Prior

- Gaussian noisy observations: $f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{x}, \sigma^2\mathbf{I})$
- Gaussian prior: $f_{\mathbf{X}}(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{C})$
- MMSE estimate:

$$\arg\min_{\hat{\mathbf{x}}} \mathbb{E}[\|\hat{\mathbf{x}} - \mathbf{X}\|_2^2|\mathbf{y}] = \mathbb{E}[\mathbf{X}|\mathbf{y}] = \big(\sigma^2\mathbf{C} + \mathbf{I}\big)^{-1}\big(\sigma^2\mathbf{C}^{-1}\boldsymbol{\mu} + \mathbf{y}\big)$$

MMSE Estimate with GMM Prior

- Gaussian noisy observations: $f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{x}, \sigma^2\mathbf{I})$
- Gaussian prior: $f_{\mathbf{X}}(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{C})$
- MMSE estimate:

$$\arg\min_{\hat{\mathbf{x}}} \mathbb{E}[\|\hat{\mathbf{x}} - \mathbf{X}\|_2^2|\mathbf{y}] = \mathbb{E}[\mathbf{X}|\mathbf{y}] = \left(\sigma^2\mathbf{C} + \mathbf{I}\right)^{-1}\left(\sigma^2\mathbf{C}^{-1}\boldsymbol{\mu} + \mathbf{y}\right)$$

- Gaussian mixture prior: $f_{\mathbf{X}}(\mathbf{x}) = \sum_{j=1}^{K} \alpha_j \, \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \mathbf{C}_j)$

MMSE Estimate with GMM Prior

- Gaussian noisy observations: $f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{x}, \sigma^2\mathbf{I})$
- Gaussian prior: $f_{\mathbf{X}}(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{C})$
- MMSE estimate:

$$\arg\min_{\hat{\mathbf{x}}} \mathbb{E}[\|\hat{\mathbf{x}} - \mathbf{X}\|_2^2|\mathbf{y}] = \mathbb{E}[\mathbf{X}|\mathbf{y}] = \left(\sigma^2\mathbf{C} + \mathbf{I}\right)^{-1}\left(\sigma^2\mathbf{C}^{-1}\boldsymbol{\mu} + \mathbf{y}\right)$$

- Gaussian mixture prior: $f_{\mathbf{X}}(\mathbf{x}) = \sum_{j=1}^{K} \alpha_j \, \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \mathbf{C}_j)$
- MMSE estimate

$$\mathbb{E}[\mathbf{X}|\mathbf{y}] = \sum_{j=1}^{K} \beta_j(\mathbf{y})\left(\sigma^2\mathbf{C}_j + \mathbf{I}\right)^{-1}\left(\sigma^2\mathbf{C}_j^{-1}\boldsymbol{\mu}_j + \mathbf{y}\right)$$

where $\beta_j(\mathbf{y}) \propto \alpha_j\mathcal{N}(\mathbf{y}|\boldsymbol{\mu}_j, \mathbf{C}_j + \sigma^2\mathbf{I})$, with $\sum_{j=1}^{K} \beta_j(\mathbf{y}) = 1$

MMSE Estimate with GMM Prior

- Gaussian noisy observations: $f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{x}, \sigma^2\mathbf{I})$
- Gaussian prior: $f_{\mathbf{X}}(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{C})$
- MMSE estimate:

$$\arg\min_{\hat{\mathbf{x}}} \mathbb{E}[\|\hat{\mathbf{x}} - \mathbf{X}\|_2^2|\mathbf{y}] = \mathbb{E}[\mathbf{X}|\mathbf{y}] = \big(\sigma^2\mathbf{C} + \mathbf{I}\big)^{-1}\big(\sigma^2\mathbf{C}^{-1}\boldsymbol{\mu} + \mathbf{y}\big)$$

- Gaussian mixture prior: $f_{\mathbf{X}}(\mathbf{x}) = \sum_{j=1}^{K} \alpha_j \, \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \mathbf{C}_j)$
- MMSE estimate (the MAP solution has no closed form)

$$\mathbb{E}[\mathbf{X}|\mathbf{y}] = \sum_{j=1}^{K} \beta_j(\mathbf{y})\big(\sigma^2\mathbf{C}_j + \mathbf{I}\big)^{-1}\big(\sigma^2\mathbf{C}_j^{-1}\boldsymbol{\mu}_j + \mathbf{y}\big)$$

where $\beta_j(\mathbf{y}) \propto \alpha_j\mathcal{N}(\mathbf{y}|\boldsymbol{\mu}_j, \mathbf{C}_j + \sigma^2\mathbf{I})$, with $\sum_{j=1}^{K} \beta_j(\mathbf{y}) = 1$

Plug-and-Play ADMM: Deblurring of Generic Images

- Generic GMM prior

Image:	Cameraman						House					
Experiment:	1	2	3	4	5	6	1	2	3	4	5	6
IDD-BM3D [Danielyan et al., 2012]	**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 [Teodoro et al., 2016]	8.39	6.36	9.80	3.47	4.16	4.88	9.66	8.22	12.43	5.50	5.42	6.82

(a) Original (b) Blurred (c) IDD-BM3D (d) ADMM-GMM

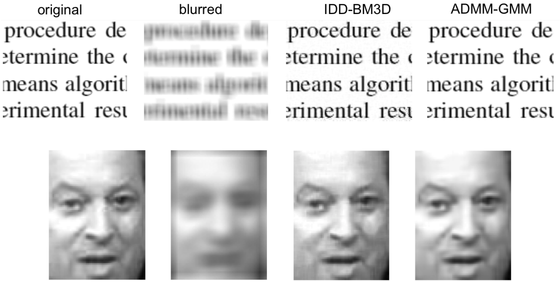
- For generic natural images: competitive, but does not beat state-of-the-art

Class-Adapted GMM-Based Restoration

- Learn a GMM for a class of images, plug the corresponding denoiser into ADMM [Teodoro et al., 2017b]

Class-Adapted GMM-Based Restoration

- Learn a GMM for a class of images, plug the corresponding denoiser into ADMM [Teodoro et al., 2017b]



Class-Adapted GMM-Based Restoration

- Learn a GMM for a class of images, plug the corresponding denoiser into ADMM [Teodoro et al., 2017b]

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 patch-based GMM-MMSE 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 patch-based GMM-MMSE denoiser

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

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

Convergence

- PnP-ADMM with a patch-based GMM-MMSE denoiser

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

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

- From Moreau [1965]: some map $p : \mathbb{R}^n \rightarrow \mathbb{R}^n$ 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.*, $\exists\, \phi : \mathbb{R}^n \rightarrow \mathbb{R} :\ p(\mathbf{x}) \in \partial\phi(\mathbf{x}),\ \forall\, \mathbf{x}$

Convergence

- PnP-ADMM with a patch-based GMM-MMSE denoiser

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

- if 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.*, $\exists\, \phi : \mathbb{R}^n \to \mathbb{R} :\ p(\mathbf{x}) \in \partial\phi(\mathbf{x}),\ \forall\, \mathbf{x}$

- Does the patch-based GMM-MMSE denoiser satisfy these conditions?

Convergence (2)

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

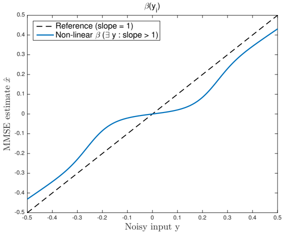
Convergence (2)

- 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 (2)

- 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:

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

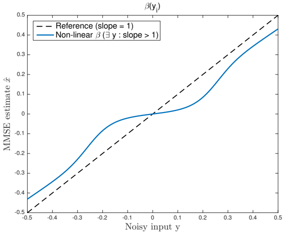


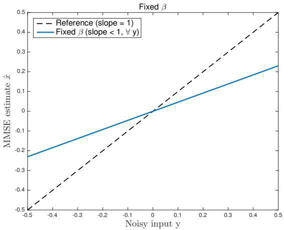
Convergence (2)

- 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 (3)

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

Convergence (3)

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

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

Convergence (3)

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

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

Convergence (3)

- 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

Convergence (3)

- 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: $\beta_m^i \leftarrow \beta_m^i / \sum_j \beta_j^i$)

Convergence (3)

- 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:

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

Convergence (3)

- 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., 2019]: for any $\sigma^2 > 0$,

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

Convergence (4)

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

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

Convergence (4)

- 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: $\mathbf{W}(\sigma^2)\mathbf{y} = \nabla_{\mathbf{y}}\big(\frac{1}{2}\mathbf{y}^T\mathbf{W}(\sigma^2)\mathbf{y}\big)$

Convergence (4)

- 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 $\lambda_{\max}\big(\mathbf{W}(\sigma^2)\big) < 1$
  - It is the gradient of a convex function: $\mathbf{W}(\sigma^2)\mathbf{y} = \nabla_{\mathbf{y}}\big(\frac{1}{2}\mathbf{y}^T\mathbf{W}(\sigma^2)\mathbf{y}\big)$

- Can we identify the function of which this denoiser is the prox?

$$\phi(\mathbf{x}) = \iota_{S(\mathbf{W})}(\mathbf{x}) + \frac{1}{2}\mathbf{x}^T\bar{\mathbf{Q}}(\bar{\Lambda}^{-1} - \mathbf{I})\bar{\mathbf{Q}}^T\mathbf{x}$$

where $S(\mathbf{W})$ is the column span of $\mathbf{W}$, $\bar{\Lambda}$ has the positive eigenvalues of $\mathbf{W}$, and $\bar{\mathbf{Q}}$ the corresponding eigenvectors.

Convergence (4)

- 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: $\mathbf{W}(\sigma^2)\mathbf{y} = \nabla_{\mathbf{y}}\big(\frac{1}{2}\mathbf{y}^T\mathbf{W}(\sigma^2)\mathbf{y}\big)$
- Can we identify the function of which this denoiser is the prox?

$$\phi(\mathbf{x}) = \iota_{S(\mathbf{W})}(\mathbf{x}) + \frac{1}{2}\mathbf{x}^T\bar{\mathbf{Q}}(\bar{\Lambda}^{-1} - \mathbf{I})\bar{\mathbf{Q}}^T\mathbf{x}$$

  where $S(\mathbf{W})$ is the column span of $\mathbf{W}$, $\bar{\Lambda}$ has the positive eigenvalues of $\mathbf{W}$, and $\bar{\mathbf{Q}}$ the corresponding eigenvectors.
- Conclusion: the problem has a solution and PnP-ADMM converges

Outline

Blind Deblurring

- Blind image deblurring/deconvolution

$$\mathbf{y} = \mathbf{h} * \mathbf{x} + \mathbf{n}$$

where both $\mathbf{x}$ and $\mathbf{h}$ are unknown

Blind Deblurring

- Blind image deblurring/deconvolution

$$\mathbf{y} = \mathbf{h} * \mathbf{x} + \mathbf{n} = \mathbf{H}(\mathbf{h})\, \mathbf{x} + \mathbf{n}$$

where both $\mathbf{x}$ and $\mathbf{h}$ are unknown

Blind Deblurring

- Blind image deblurring/deconvolution

$$\mathbf{y} = \mathbf{h} * \mathbf{x} + \mathbf{n} = \mathbf{H}(\mathbf{h})\, \mathbf{x} + \mathbf{n} = \mathbf{X}(\mathbf{x})\, \mathbf{h} + \mathbf{n}$$

where both $\mathbf{x}$ and $\mathbf{h}$ are unknown

Blind Deblurring

- Blind image deblurring/deconvolution

$$\mathbf{y} = \mathbf{h} * \mathbf{x} + \mathbf{n} = \mathbf{H}(\mathbf{h})\, \mathbf{x} + \mathbf{n} = \mathbf{X}(\mathbf{x})\, \mathbf{h} + \mathbf{n}$$

  where both $\mathbf{x}$ and $\mathbf{h}$ are unknown

- Joint criterion (under Gaussian noise) [Almeida and Figueiredo, 2013b]

$$(\hat{\mathbf{x}}, \hat{\mathbf{h}}) \in \arg\min_{\mathbf{x},\mathbf{h}} \underbrace{\frac{1}{2}\|\mathbf{h} * \mathbf{x} - \mathbf{y}\|_2^2 + \Phi(\mathbf{x}) + \Psi(\mathbf{h})}_{O(\mathbf{x},\mathbf{h})}$$

  where Φ and Ψ are regularizers

Blind Deblurring

- Blind image deblurring/deconvolution

$$\mathbf{y} = \mathbf{h} * \mathbf{x} + \mathbf{n} = \mathbf{H}(\mathbf{h})\,\mathbf{x} + \mathbf{n} = \mathbf{X}(\mathbf{x})\,\mathbf{h} + \mathbf{n}$$

  where both $\mathbf{x}$ and $\mathbf{h}$ are unknown

- Joint criterion (under Gaussian noise) [Almeida and Figueiredo, 2013b]

$$(\hat{\mathbf{x}}, \hat{\mathbf{h}}) \in \arg\min_{\mathbf{x},\mathbf{h}} \underbrace{\frac{1}{2}\|\mathbf{h} * \mathbf{x} - \mathbf{y}\|_2^2 + \Phi(\mathbf{x}) + \Psi(\mathbf{h})}_{O(\mathbf{x},\mathbf{h})}$$

  where Φ and Ψ are regularizers

- Even if Φ and Ψ are convex, this is a non-convex problem

Algorithm

- Proximal alternating minimization [Attouch et al., 2007]

Algorithm

- Proximal alternating minimization [Attouch et al., 2007]
- Solver for each minimization: ADMM/SALSA

Algorithm

- Proximal alternating minimization [Attouch et al., 2007]
- Solver for each minimization: ADMM/SALSA

Initialization: $\hat{\mathbf{x}} = \mathbf{y}$, $\hat{\mathbf{h}}$ - identity filter
while stopping criterion is not satisfied **do**

$$\hat{\mathbf{x}} \leftarrow \arg\min_{\mathbf{x}} \; O(\mathbf{x}, \hat{\mathbf{h}}) + \frac{\rho_x}{2}\|\mathbf{x} - \hat{\mathbf{x}}^{\text{previous}}\|^2$$

$$\hat{\mathbf{h}} \leftarrow \arg\min_{\mathbf{h}} \; O(\hat{\mathbf{x}}, \mathbf{h}) + \frac{\rho_h}{2}\|\mathbf{h} - \hat{\mathbf{h}}^{\text{previous}}\|^2$$

end while

Algorithm

- Proximal alternating minimization [Attouch et al., 2007]
- Solver for each minimization: ADMM/SALSA

Initialization: $\hat{\mathbf{x}} = \mathbf{y}$, $\hat{\mathbf{h}}$ - identity filter
while stopping criterion is not satisfied **do**
  $\hat{\mathbf{x}} \leftarrow \underset{\mathbf{x}}{\text{argmin}}\; O(\mathbf{x}, \hat{\mathbf{h}}) + \frac{\rho_x}{2}\|\mathbf{x} - \hat{\mathbf{x}}^{\text{previous}}\|^2$
  $\hat{\mathbf{h}} \leftarrow \underset{\mathbf{h}}{\text{argmin}}\; O(\hat{\mathbf{x}}, \mathbf{h}) + \frac{\rho_h}{2}\|\mathbf{h} - \hat{\mathbf{h}}^{\text{previous}}\|^2$
end while

- Image regularizer: class-adapted plug-and-play priors

Algorithm

- Proximal alternating minimization [Attouch et al., 2007]
- Solver for each minimization: ADMM/SALSA

Initialization: $\hat{\mathbf{x}} = \mathbf{y}$, $\hat{\mathbf{h}}$ - identity filter
while stopping criterion is not satisfied **do**

$$\hat{\mathbf{x}} \leftarrow \underset{\mathbf{x}}{\text{argmin}}\ O(\mathbf{x}, \hat{\mathbf{h}}) + \frac{\rho_x}{2}\|\mathbf{x} - \hat{\mathbf{x}}^{\text{previous}}\|^2$$

$$\hat{\mathbf{h}} \leftarrow \underset{\mathbf{h}}{\text{argmin}}\ O(\hat{\mathbf{x}}, \mathbf{h}) + \frac{\rho_h}{2}\|\mathbf{h} - \hat{\mathbf{h}}^{\text{previous}}\|^2$$

end while

- Image regularizer: class-adapted plug-and-play priors
- Filter regularizer: positivity and support, or sparsity

Priors

- Plug-and-play image priors:

Priors

- Plug-and-play image priors:
  - ✓ **GMM**-based patch denoiser, trained on a dataset of clean images (text, faces, fingerprint)

Priors

- Plug-and-play image priors:
  - ✓ **GMM**-based patch denoiser, trained on a dataset of clean images (text, faces, fingerprint)
  - ✓ **Dictionary**-based patch denoiser, learned from clean images (same classes)

Priors

- Plug-and-play image priors:
  - ✓ **GMM**-based patch denoiser, trained on a dataset of clean images (text, faces, fingerprint)
  - ✓ **Dictionary**-based patch denoiser, learned from clean images (same classes)
  - ✓ General-purpose **BM3D** denoiser.

Priors

- Plug-and-play image priors:
  - ✓ **GMM**-based patch denoiser, trained on a dataset of clean images (text, faces, fingerprint)
  - ✓ **Dictionary**-based patch denoiser, learned from clean images (same classes)
  - ✓ General-purpose **BM3D** denoiser.
- Blur filter priors

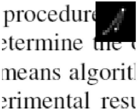
Priors

- Plug-and-play image priors:
  - ✓ **GMM**-based patch denoiser, trained on a dataset of clean images (text, faces, fingerprint)
  - ✓ **Dictionary**-based patch denoiser, learned from clean images (same classes)
  - ✓ General-purpose **BM3D** denoiser.
- Blur filter priors
  - ✓ Constraint: positivity and maximum support

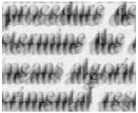
Priors

- Plug-and-play image priors:
  - ✓ **GMM**-based patch denoiser, trained on a dataset of clean images (text, faces, fingerprint)
  - ✓ **Dictionary**-based patch denoiser, learned from clean images (same classes)
  - ✓ General-purpose **BM3D** denoiser.
- Blur filter priors
  - ✓ Constraint: positivity and maximum support
  - ✓ Sparsity (adequate for motion blur)

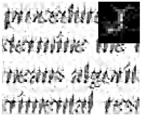
Results: GMM-based prior for text images



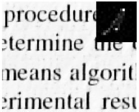
original



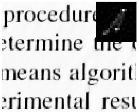
blurred



[Pan et al., 2014]



BM3D: 9.97 dB



GMM: 11.16 dB

Experiments

original

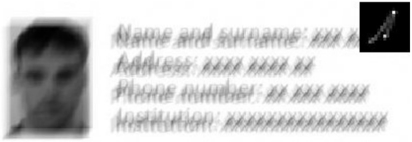
blurred

[Almeida and Figueiredo, 2013b]
0.36 dB

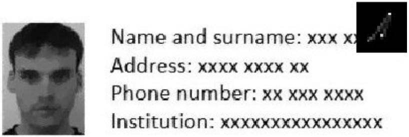
BM3D: 0.66 dB

GMM: 1.19 dB

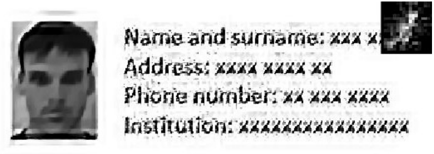
Experiments



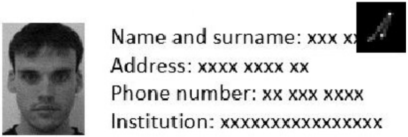
(a) Blurred image



(b) [Almeida and Figueiredo, 2013b]

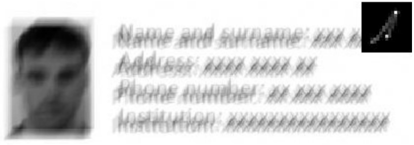


(c) [Pan et al., 2014]

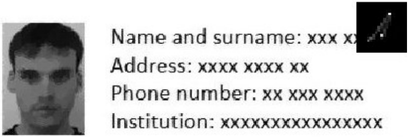


(d) Proposed

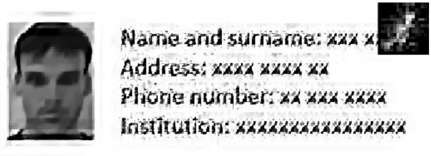
Experiments



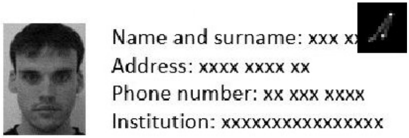
(a) Blurred image



(b) [Almeida and Figueiredo, 2013b]



(c) [Pan et al., 2014]

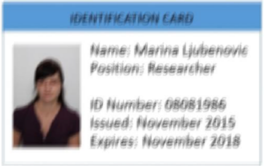


(d) Proposed

- Uses a concatenation of two dictionaries: face and text

Experiments

original



blurred

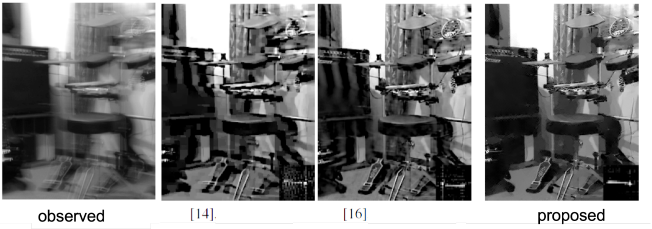
[Krishnan et al, 2011]

[Xu and Jia, 2011]

[Pan et al, 2014]

proposed

Blind Deconvolution: Real Examples

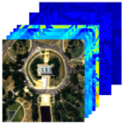


Results from [Almeida and Figueiredo, 2013b]

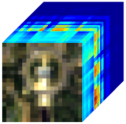
Outline

An Extreme Case of Adaptation: Hyperspectral Fusion

- Spectral-spatial resolution trade-off:



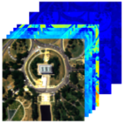
Multi-spectral:
high spatial resolution
low spectral resolution



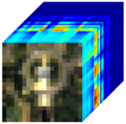
Hyper-spectral:
low spatial resolution
high spectral resolution

An Extreme Case of Adaptation: Hyperspectral Fusion

- Spectral-spatial resolution trade-off:



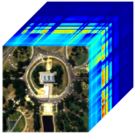
Multi-spectral:
high spatial resolution
low spectral resolution



Hyper-spectral:
low spatial resolution
high spectral resolution

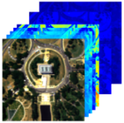
- Fuse MS and HS data:

  high spatial & spectral resolutions

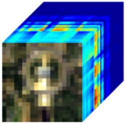


An Extreme Case of Adaptation: Hyperspectral Fusion

- Spectral-spatial resolution trade-off:



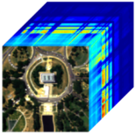
Multi-spectral:
high spatial resolution
low spectral resolution



Hyper-spectral:
low spatial resolution
high spectral resolution

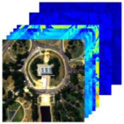
- Fuse MS and HS data:

  high spatial & spectral resolutions

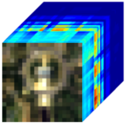


- Extreme case: pansharpening (panchromatic rather than MS image).

- Spectral-spatial resolution trade-off:



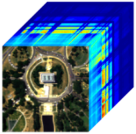
Multi-spectral:
high spatial resolution
low spectral resolution



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., 2015]

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

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 $n_h < n_m$

Hyperspectral Fusion: Formulation

- Observation model [Simões et al., 2015]

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

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

$$L_h > L_m \text{ and } n_h < n_m$$

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

Hyperspectral Fusion: Formulation

- Observation model [Simões et al., 2015]

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

$L_h > L_m$ and $n_h < n_m$

✓ $\mathbf{E} \in \mathbb{R}^{L_h \times p}$: 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., 2015]

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

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

$L_h > L_m$ and $n_h < n_m$

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

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

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

Hyperspectral Fusion: Formulation

- Observation model [Simões et al., 2015]

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

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 $n_h < n_m$

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

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

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

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

Hyperspectral Fusion: Formulation

- Observation model [Simões et al., 2015]

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

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 $n_h < n_m$

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

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

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

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

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

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:

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

- ...which fits nicely the SALSA template ($J = 3$): $\quad \min_{\mathbf{x}} \sum_{j=1}^{J} g_j\big(\mathbf{H}_j\, \mathbf{x}\big)$

Hyperspectral Fusion via PnP-ADMM

- Assuming Gaussian noise:

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

- ...which fits nicely the SALSA template ($J = 3$): $\min_{\mathbf{x}} \sum_{j=1}^{J} g_j\big(\mathbf{H}_j \, \mathbf{x}\big)$

- Matrix inversion computable via FFT (with periodic or unknown BC)

Hyperspectral Fusion via PnP-ADMM

- Assuming Gaussian noise:

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

- ...which fits nicely the SALSA template ($J = 3$): $\quad \min_{\mathbf{x}} \sum_{j=1}^{J} g_j\big(\mathbf{H}_j\, \mathbf{x}\big)$

- Matrix inversion computable via FFT (with periodic or unknown BC)

- Proximity operators:
  - ✓ The one involving $\mathbf{RE}$: a single $p \times p$ inversion; decoupled across pixels

Hyperspectral Fusion via PnP-ADMM

- Assuming Gaussian noise:

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

- ...which fits nicely the SALSA template ($J = 3$): $\displaystyle \min_{\mathbf{x}} \sum_{j=1}^{J} g_j\big(\mathbf{H}_j\, \mathbf{x}\big)$

- Matrix inversion computable via FFT (with periodic or unknown BC)

- Proximity operators:
  - ✓ The one involving $\mathbf{RE}$: a single $p \times p$ inversion; decoupled across pixels
  - ✓ The one involving $\mathbf{BM}$: solved by FFT, decoupled across bands

Hyperspectral Fusion via PnP-ADMM

- Assuming Gaussian noise:

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

- ...which fits nicely the SALSA template ($J = 3$): $\quad \min_{\mathbf{x}} \sum_{j=1}^{J} g_j\big(\mathbf{H}_j\,\mathbf{x}\big)$

- Matrix inversion computable via FFT (with periodic or unknown BC)

- Proximity operators:
  - ✓ The one involving $\mathbf{RE}$: a single $p \times p$ inversion; decoupled across pixels
  - ✓ The one involving $\mathbf{BM}$: solved by FFT, decoupled across bands
  - ✓ The prox of ϕ is replaced by an adapted GMM-based denoiser

Hyperspectral Fusion via PnP-ADMM

- Assuming Gaussian noise:

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

- ...which fits nicely the SALSA template ($J = 3$): $\min_{\mathbf{x}} \sum_{j=1}^{J} g_j\big(\mathbf{H}_j\, \mathbf{x}\big)$
- Matrix inversion computable via FFT (with periodic or unknown BC)
- Proximity operators:
  - ✓ The one involving $\mathbf{RE}$: a single $p \times p$ inversion; decoupled across pixels
  - ✓ The one involving $\mathbf{BM}$: solved by FFT, decoupled across bands
  - ✓ The prox of ϕ is replaced by an adapted GMM-based denoiser
- The GMM is learned from patches of $\mathbf{Y}_m$ (high spatial resolution) [Teodoro et al., 2019]

Hyperspectral Fusion: Synthetic Example

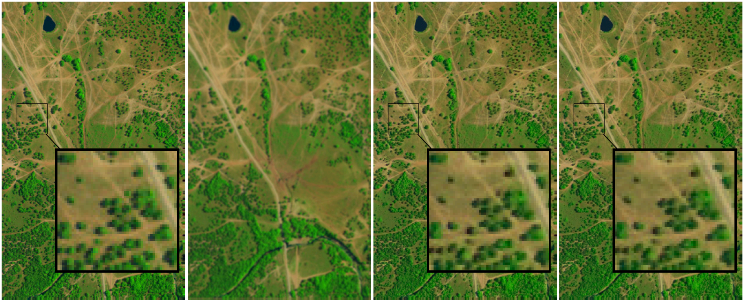
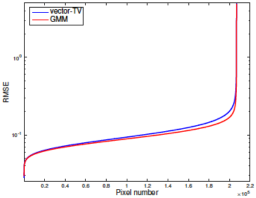


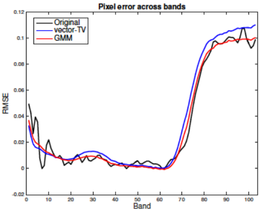
Table 3: HS and MS fusion on RTerrain dataset.

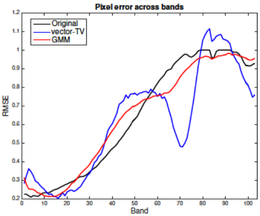
	Exp. 1 (PAN)			Exp. 2 (PAN)			Exp. 3 (R,G,B,N-IR)			Exp. 4 (R,G,B,N-IR)		
SNR ($\mathbf{Y}_m$)	50dB			30dB			50dB			30dB		
SNR ($\mathbf{Y}_h$)	50dB			20dB			50dB			20dB		
Metric	ERGAS	SAM	SRE	ERGAS	SAM	SRE	ERGAS	SAM	SRE	ERGAS	SAM	SRE
HySure	2.62	5.34	21.46	2.77	5.35	20.86	1.08	2.68	28.71	1.53	3.42	26.07
Proposed	2.58	**5.15**	**21.69**	**2.75**	**5.33**	**21.12**	**0.91**	**2.20**	**30.86**	**1.29**	**2.85**	**27.85**
ADMM-BM3D	**2.57**	5.17	21.65	2.76	5.36	21.08	0.93	2.22	30.80	1.31	2.91	27.72

[Teodoro et al., 2017a]

Hyperspectral Fusion: Synthetic Example







Outline

Final Remarks

- ADMM/SALSA: a flexible toolbox for a variety inverse problems

Final Remarks

- ADMM/SALSA: a flexible toolbox for a variety inverse problems

- Its speed hinges on the inversion of $(\mathbf{B}^T\mathbf{B} + \mathbf{I})$ (à la quasi-Newton)

Final Remarks

- ADMM/SALSA: a flexible toolbox for a variety inverse problems

- Its speed hinges on the inversion of $(\mathbf{B}^T\mathbf{B} + \mathbf{I})$ (à la quasi-Newton)

- Plug-and-play (PnP) denoisers "can" be used with ADMM

Final Remarks

- ADMM/SALSA: a flexible toolbox for a variety inverse problems
- Its speed hinges on the inversion of $(\mathbf{B}^T\mathbf{B} + \mathbf{I})$ (à la quasi-Newton)
- Plug-and-play (PnP) denoisers "can" be used with ADMM
- Convergence properties of PnP-ADMM with fixed linear denoiser

Final Remarks

- ADMM/SALSA: a flexible toolbox for a variety inverse problems
- Its speed hinges on the inversion of $(\mathbf{B}^T\mathbf{B} + \mathbf{I})$ (à la quasi-Newton)
- Plug-and-play (PnP) denoisers "can" be used with ADMM
- Convergence properties of PnP-ADMM with fixed linear denoiser
- Extension to blind deblurring (non-convex)

Final Remarks

- ADMM/SALSA: a flexible toolbox for a variety inverse problems
- Its speed hinges on the inversion of $(\mathbf{B}^T\mathbf{B} + \mathbf{I})$ (à la quasi-Newton)
- Plug-and-play (PnP) denoisers "can" be used with ADMM
- Convergence properties of PnP-ADMM with fixed linear denoiser
- Extension to blind deblurring (non-convex)
- Ideally suited for hyperspectral imaging

Thank you.

References I

..., ...

M.V. Afonso, J.M. Bioucas-Dias, and M.A.T. Figueiredo. An augmented lagrangian approach to the constrained optimization formulation of imaging inverse problems. *IEEE Transactions on Image Processing*, 20:681–695, 2011.

M. Almeida and M. Figueiredo. Deconvolving images with unknown boundaries using the alternating direction method of multipliers. *IEEE Transactions on Image Processing*, 22: 3074–3086, 2013a.

M. Almeida and M. Figueiredo. Blind image deblurring with unknown boundary conditions using the alternating direction method of multipliers. In *IEEE International Conf. on Image Processing*, 2013b.

H. Attouch, P. Redont, and A. Soubeyran. A new class of alternating proximal minimization algorithms with costs to move. *SIAM Journal on Optimization*, 18:1061–1081, 2007.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3:1–122, 2011.

S. Chan, X. Wang, and O. Elgendy. Plug-and-play ADMM for image restoration: Fixed point convergence and applications. *IEEE Transactions on Computational Imaging*, 3:in press, 2017.

T. Chan, A. Yip, and F. Park. Simultaneous total variation image inpainting and blind deconvolution. *International Journal of Imaging Systems Technology*, 15:92–102, 2005.

P.L. Combettes and J.C. Pesquet. Proximal splitting methods in signal processing. *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212, 2011.

References II

K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16: 2080–2095, 2007.

A. Danielyan, V. Katkovnik, and K. Egiazarian. BM3D frames and variational image deblurring. *IEEE Transactions on Image Processing*, 21:1715–1728, 2012.

J. Eckstein and D. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 5:293–318, 1992.

M. Figueiredo and J. Bioucas-Dias. Restoration of Poissonian images using alternating direction optimization. *IEEE Transactions on Image Processing*, 19:3133–3145, 2010.

D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximations. *Computers and Mathematics with Applications*, 2:17–40, 1976.

R. Glowinski and A. Marrocco. Sur l'approximation, par elements finis d'ordre un, et la resolution, par penalisation-dualité, d'une classe de problemes de Dirichlet non lineares. *Revue Française d'Automatique, Informatique et Recherche Opérationelle*, 9:41–76, 1975.

Reinhard Heckel and Paul Hand. Deep decoder: Concise image representations from untrained non-convolutional networks. *arXiv:1810.03982*, 2019.

M. Lebrun, A. Buades, and J. M. Morel. A nonlocal Bayesian image denoising algorithm. *SIAM Journal on Imaging Science*, 6:1665–1688, 2013.

J. J. Moreau. Proximité et dualtité dans un espace hilbertien. *Bulletin de la Société Mathématique de France*, 93:273–299, 1965.

References III

J. Pan, Z. Hu, Z. Su, and M. Yang. Deblurring text images via ℓ_0-regularized intensity and gradient prior. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

A. Matakos S. Ramani and J. Fessler. Accelerated edge-preserving image restoration without boundary artifacts. *IEEE Transactions on Image Processing*, 22:2019–2029, 2013.

S. Reeves. Fast image restoration without boundary artifacts. *IEEE Transactions on Image Processing*, 14:1448–1453, 2005.

S. Setzer, G. Steidl, and T. Teuber. Deblurring poissonian images by split Bregman techniques. *Journal of Visual Communication and Image Representation*, 21:193–199, 2010.

M. Simões, J. Bioucas-Dias, L. Almeida, and J. Chanussot. A convex formulation for hyperspectral image superresolution via subspace-based regularization. *IEEE Transactions on Geoscience and Remote Sensing*, 55:3373–3388, 2015.

S. Sreehari, S. Venkatakrishnan, B. Wohlberg, G. Buzzard, L. Drummy, J. Simmons, and A. Bouman. Plug-and-play priors for bright field electron tomography and sparse interpolation. *IEEE Transactions on Computational Imaging*, 2:408–423, 2016.

A. Teodoro, M. Almeida, and M. Figueiredo. Single-frame image denoising and inpainting using Gaussian mixtures. In *4th International Conference on Pattern Recognition Applications and Methods*, 2015.

A. Teodoro, J. Bioucas-Dias, and M. Figueiredo. Image restoration and reconstruction using variable splitting and class-adapted image priors. In *IEEE International Conference on Image Processing*, 2016.

References IV

A. Teodoro, J. Bioucas-Dias, and M. Figueiredo. Image restoration with locally selected class-adapted models. In *IEEE 26th International Workshop on Machine Learning for Signal Processing*, 2017a.

A. Teodoro, J. Bioucas-Dias, and M. Figueiredo. Scene-adapted plug-and-play with convergence guarantees. In *IEEE International Workshop on Machine Learning for Signal Processing*, 2017b.

A. Teodoro, J. Bioucas-Dias, and M. Figueiredo. A convergent image fusion algorithm using scene-adapted Gaussian-mixture-based denoising, 2019. IEEE Transactions on Image Processing.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. *arXiv:1711.10925*, 2017.

S. Venkatakrishnan, C. Bouman, E. Chu, and B. Wohlberg. Plug-and-play priors for model based reconstruction. In *IEEE Global Conference on Signal and Information Processing*, pages 945–948, 2013.

X. Xu, Y. Sun, J. Liu, B. Wohlberg, and U. Kamilov. Provable convergence of Plug-and-Play priors with MMSE denoisers. *IEEE Signal Processing Letters*, 27:1280–1284, 2020.

Z. Xu, M. Figueiredo, and T. Goldstein. Adaptive ADMM with spectral penalty parameter selection. In *Artificial Intelligence and Statistics (AISTATS)*, 2016.

Z. Xu, M. Figueiredo, X. Yuan, C. Studer, and T. Goldstein. Adaptive relaxed ADMM: Convergence theory and practical implementation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017a.

References V

Z. Xu, G. Taylor, H. Li, Figueiredo, X. Yuan, and T. Goldstein. Adaptive consensus ADMM for distributed optimization. In *International Conference on Machine Learning (ICML)*, 2017b.

D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *International Conference on Computer Vision*, pages 479–486, 2011.