On the Use of Alternating Direction Optimization for Imaging Inverse Problems

Mário A. T. Figueiredo

Instituto de Telecomunicações and Instituto Superior Técnico, Technical University of Lisbon

PORTUGAL

mario.figueiredo@lx.it.pt, www.lx.it.pt/~mtf

Joint work with José Bioucas-Dias and Manya Afonso
Outline

1. Review of some classical imaging inverse problems
2. Alternating direction method of multipliers (ADMM) …for sums of two or more functions.
3. Linear/Gaussian image reconstruction/restoration: SALSA
4. Deblurring Poissonian images: PIDAL
5. Other applications: structured sparsity, hybrid regularization, …
Regularized Solution of Inverse Problems

Many ill-posed inverse problems are addressed by solving

\[ \hat{x} \in \arg \min_{x \in \mathbb{R}^n} f(x) + \tau c(x) \]

\( f : \mathbb{R}^n \rightarrow \mathbb{R} \) data fidelity, observation model, negative log-likelihood, …
usually smooth and convex.

\( c : \mathbb{R}^n \rightarrow \mathbb{R} \) regularization/penalty function, or negative log-prior;
typically convex non-differentiable (e.g., for sparsity).

Examples: frame-based signal/image restoration/reconstruction,
sparse representations, compressive sensing, …
A Fundamental Dichotomy: Analysis vs Synthesis
[Elad, Milanfar, Rubinstein, 2007], [Selesnick, F, 2010]

$$\hat{x} \in \arg \min_x \mathcal{L}(Ax) + \tau c(x)$$

Frame-based “synthesis” regularization

\(x\) contains representation coefficients (not the signal/image itself)

\(A = BW\), where \(B\) is the observation operator

\(W\) is a synthesis operator (e.g. of a Parseval frame)

\(WW^* = I\)

typical (sparseness-inducing) regularizer

\(c(x) = \|x\|_1\)

proper, lsc, convex (not strictly), and coercive.
A Fundamental Dichotomy: Analysis vs Synthesis

\[ \hat{x} \in \operatorname*{arg\,min}_{x} \mathcal{L}(Ax) + \tau c(x) \]

Frame-based “analysis” regularization

\[ x \text{ is the signal/image itself, } A \text{ is the observation operator} \]

typical frame-based analysis regularizer:

\[ c(x) = \|Px\|_1 \]

analysis operator (e.g., of a Parseval frame)

\[ P^*P = I \]

proper, lsc, convex (not strictly), and coercive.
**Image Restoration/Reconstruction: General Formulation**

All the previous models written as

\[ f(x) = \mathcal{L}(Ax) \]

\[ \mathcal{L}(z) = \sum_{i=1}^{m} \xi(z_i, y_i) \]

where \( \xi \) is one (e.g.) of these functions:

**Gaussian observations:**

\[ \xi_G(z, y) = \frac{1}{2} (z - y)^2 \quad \rightarrow \quad \mathcal{L}_G \]

**Poissonian observations:**

\[ \xi_P(z, y) = z + \nu_{\mathbb{R}_+}(z) - y \log(z) \quad \rightarrow \quad \mathcal{L}_P \]

**Multiplicative noise:**

\[ \xi_M(z, y) = L(z + e^{y-z}) \quad \rightarrow \quad \mathcal{L}_M \]

...all proper, lower semi-continuous (lsc), coercive, convex.

\( \mathcal{L}_G \) and \( \mathcal{L}_M \) are strictly convex. \( \mathcal{L}_P \) is strictly convex if \( y_i > 0, \forall i \)
Proximity Operators and Iterative Shrinkage/Thresholding

\[ \hat{x} \in \arg \min_x f(x) + \tau c(x) \]

The so-called shrinkage/thresholding/denoising function,

\[ \text{prox}_{\tau c}(u) = \arg \min_x \frac{1}{2} \| x - u \|^2_2 + \tau c(x) \]

or Moreau proximity operator [Moreau 62], [Combettes 01], [Combettes, Wajs, 05].

Classical case: \( c(z) = \| z \|_1 \Rightarrow \text{prox}_{\tau c}(u) = \text{soft}(u, \tau) \)

**IST algorithm:**
[F and Nowak, 01, 03],
[Daubechies, Defrise, De Mol, 02, 04],
[Combettes and Wajs, 03, 05],
[Starck, Candés, Nguyen, Murtagh, 2003],

\[ x_{k+1} = \text{prox}_{\tau c/\alpha} \left( x_k - \frac{1}{\alpha} \nabla f(x_k) \right) \]

**Forward-backward splitting**
[Bruck, 1977], [Passty, 1979], [Lions and Mercier, 1979],
Drawbacks of IST

\[ x_{k+1} = \text{prox}_{\tau c/\alpha} \left( x_k - \frac{1}{\alpha} \nabla f(x_k) \right) \]

Key condition in convergence proofs: \( \nabla f \) is Lipschitz

...not true with Poisson or multiplicative noise.

Even for the linear/Gaussian case
\[ f(x) = \frac{1}{2} \| A x - y \|^2 \]

...IST is known to be slow when \( A \) is ill-conditioned and/or when \( \tau \) is very small.

Accelerated versions of IST: Two-step IST (TwIST) [Bioucas-Dias, F, 07]

Fast IST (FISTA) [Beck and Teboulle, 09]

Fixed-point continuation (FPC) [Hale, Yin, and Zhang, 07]

GPSR [F, Nowak, Wright, 07]

SpaRSA [Wright, Nowak, F, 08, 09]

several others…
ADMM: Variable Splitting + Augmented Lagrangian View

Unconstrained (convex) optimization problem:
\[ \min_{\mathbf{z} \in \mathbb{R}^d} f_1(\mathbf{z}) + f_2(\mathbf{G} \mathbf{z}) \]

Equivalent constrained problem:
\[ \min_{\mathbf{z} \in \mathbb{R}^d, \mathbf{u} \in \mathbb{R}^c} f_1(\mathbf{z}) + f_2(\mathbf{u}) \]
\[ \text{s.t. } \mathbf{u} - \mathbf{G} \mathbf{z} = \mathbf{0} \]

Augmented Lagrangian (AL):
\[ L_\mu(\mathbf{z}, \mathbf{u}, \lambda) = f_1(\mathbf{z}) + f_2(\mathbf{u}) + \lambda^T(\mathbf{G} \mathbf{z} - \mathbf{u}) + \frac{\mu}{2} \| \mathbf{G} \mathbf{z} - \mathbf{u} \|^2 \]

AL method, or method of multipliers (MM) [Hestenes, Powell, 1969]

\[ (\mathbf{z}_{k+1}, \mathbf{u}_{k+1}) = \arg \min_{\mathbf{z}, \mathbf{u}} L_\mu(\mathbf{z}, \mathbf{u}, \lambda_k) \]
\[ \lambda_{k+1} = \lambda_k + \mu(\mathbf{G} \mathbf{z}_{k+1} - \mathbf{u}_{k+1}) \]

equivalent

ADMM corresponds to minimizing alternatingly w.r.t. \( \mathbf{z} \) and \( \mathbf{u} \)

\[ (\mathbf{z}_{k+1}, \mathbf{u}_{k+1}) = \arg \min_{\mathbf{z}, \mathbf{u}} f_1(\mathbf{z}) + f_2(\mathbf{u}) + \frac{\mu}{2} \| \mathbf{G} \mathbf{z} - \mathbf{u} - \mathbf{d}_k \|^2 \]
\[ \mathbf{d}_{k+1} = \mathbf{d}_k - (\mathbf{G} \mathbf{z}_{k+1} - \mathbf{u}_{k+1}) \]
Alternating Direction Method of Multipliers (ADMM)

Unconstrained (convex) optimization problem:

$$\min_{z \in \mathbb{R}^d} f_1(z) + f_2(Gz)$$

ADMM  [Glowinski, Marrocco, 75], [Gabay, Mercier, 76]

$$z_{k+1} = \arg \min_z f_1(z) + \frac{\mu}{2} \|Gz - u_k - d_k\|^2$$

$$u_{k+1} = \arg \min_u f_2(u) + \frac{\mu}{2} \|Gz_{k+1} - u - d_k\|^2$$

$$d_{k+1} = d_k - (Gz_{k+1} - u_{k+1})$$

Interpretations: variable splitting + augmented Lagrangian + NLBGS;

Douglas-Rachford splitting on the dual  [Eckstein, Bertsekas, 92];

split-Bregman approach [Goldstein, Osher, 08]
Consider the problem

$$\min_{z \in \mathbb{R}^d} f_1(z) + f_2(Gz)$$

Let $f_1$ and $f_2$ be closed, proper, and convex and $G$ have full column rank.

Let $(z_k, \ k = 0, 1, 2, \ldots)$ be the sequence produced by ADMM, with $\mu > 0$; then, if the problem has a solution, say $\bar{z}$, then

$$\lim_{k \to \infty} z_k = \bar{u}$$

The theorem also allows for inexact minimizations, as long as the errors are absolutely summable.
ADMM for Two or More Functions

Consider a more general problem

\[
\min_{\mathbf{z} \in \mathbb{R}^d} \sum_{j=1}^{J} g_j(\mathbf{H}^{(j)} \mathbf{z}) \quad (P)
\]

Proper, closed, convex functions

\[g_j : \mathbb{R}^{p_j} \rightarrow \mathbb{R}\]

Arbitrary matrices

\[\mathbf{H}^{(j)} \in \mathbb{R}^{p_j \times d}\]

There are many ways to write \((P)\) as

\[
\min_{\mathbf{z} \in \mathbb{R}^d} f_1(\mathbf{z}) + f_2(\mathbf{G} \mathbf{z})
\]

We adopt:

\[
f_1(\mathbf{z}) = 0, \quad \mathbf{G} = \begin{bmatrix} \mathbf{H}^{(1)} \\ \vdots \\ \mathbf{H}^{(J)} \end{bmatrix}, \quad f_2 \left( \begin{bmatrix} \mathbf{u}^{(1)} \\ \vdots \\ \mathbf{u}^{(J)} \end{bmatrix} \right) = \sum_{j=1}^{J} g_j(\mathbf{u}^{(j)})
\]

Another approach in [Goldfarb, Ma, 09, 11]
Applying ADMM to More Than Two Functions

\[
\min_{z \in \mathbb{R}^d} \sum_{j=1}^{J} g_j(H^{(j)} z) \quad \min_{z \in \mathbb{R}^d} f_2(G z) \quad G = \begin{bmatrix} H^{(1)} \\ \vdots \\ H^{(J)} \end{bmatrix} \quad u = \begin{bmatrix} u^{(1)} \\ \vdots \\ u^{(J)} \end{bmatrix}
\]

\[
z_{k+1} = \arg\min_z f_1(z) + \frac{\mu}{2} \| G z - u_k - d_k \|^2
\]

\[
u_{k+1} = \arg\min_u f_2(u) + \frac{\mu}{2} \| G z_{k+1} - u - d_k \|^2
\]

\[
d_{k+1} = d_k - (G z_{k+1} - u_{k+1})
\]
Applying ADMM to More Than Two Functions

\[
\min_{z \in \mathbb{R}^d} \sum_{j=1}^J g_j(H^{(j)} z) \quad \min_{z \in \mathbb{R}^d} f_2(G z) \quad G = \begin{bmatrix} H^{(1)} \\ \vdots \\ H^{(J)} \end{bmatrix} \quad u = \begin{bmatrix} u^{(1)} \\ \vdots \\ u^{(J)} \end{bmatrix}
\]

\[
z_{k+1} = \left[ \sum_{j=1}^J (H^{(j)})^T H^{(j)} \right]^{-1} \left( \sum_{j=1}^J H^{(j)} \left( u_k^{(j)} + d_k^{(j)} \right) \right)
\]

\[
u_{k+1} = \arg \min_u f_2(u) + \frac{\mu}{2} \| G z_{k+1} - u - d_k \|^2
\]

\[d_{k+1} = d_k - (G z_{k+1} - u_{k+1})\]
Applying ADMM to More Than Two Functions

\[
\min_{z \in \mathbb{R}^d} \sum_{j=1}^{J} g_j(H^{(j)} z) \quad \min_{z \in \mathbb{R}^d} f_2(G z) \quad G = \begin{bmatrix} H^{(1)} \\ \vdots \\ H^{(J)} \end{bmatrix} \quad u = \begin{bmatrix} u^{(1)} \\ \vdots \\ u^{(J)} \end{bmatrix}
\]

\[
\begin{align*}
\mathbf{z}_{k+1} &= \left( \sum_{j=1}^{J} (H^{(j)})^T H^{(j)} \right)^{-1} \left( \sum_{j=1}^{J} H^{(j)} (u_k^{(j)} + d_k^{(j)}) \right) \\
\mathbf{u}_{k+1}^{(1)} &= \arg \min_{u} g_1(u) + \frac{\mu}{2} \| u - H^{(1)} \mathbf{z}_{k+1} + d_k^{(1)} \|^2 \\
\mathbf{u}_{k+1} = \begin{bmatrix} u_{k+1}^{(1)} \\ \vdots \\ u_{k+1}^{(J)} \end{bmatrix} \\
\mathbf{d}_{k+1} &= \mathbf{d}_k - (G \mathbf{z}_{k+1} - \mathbf{u}_{k+1})
\end{align*}
\]
Applying ADMM to More Than Two Functions

\[
\min_{z \in \mathbb{R}^d} \sum_{j=1}^{J} g_j(H^{(j)} z) \quad \min_{z \in \mathbb{R}^d} f_2(G z) \quad G = \begin{bmatrix} H^{(1)} \\ \vdots \\ H^{(J)} \end{bmatrix} \quad u = \begin{bmatrix} u^{(1)} \\ \vdots \\ u^{(J)} \end{bmatrix}
\]

\[
z_{k+1} = \left[ \sum_{j=1}^{J} (H^{(j)})^T H^{(j)} \right]^{-1} \left( \sum_{j=1}^{J} H^{(j)} \left( u_k^{(j)} + d_k^{(j)} \right) \right)
\]

\[
u_{k+1}^{(1)} = \arg \min_{u} g_1(u) + \frac{\mu}{2} \| u - H^{(1)} z_{k+1} + d_k^{(1)} \|^2 = \text{prox}_{g_1/\mu}(H^{(1)} z_{k+1} - d_k^{(1)})
\]

\[
\vdots \\
\vdots \\
\vdots \\
\vdots \\

\nu_{k+1}^{(J)} = \arg \min_{u} g_J(u) + \frac{\mu}{2} \| u - H^{(J)} z_{k+1} + d_k^{(J)} \|^2 = \text{prox}_{g_J/\mu}(H^{(J)} z_{k+1} - d_k^{(J)})
\]

\[
d_{k+1} = d_k - (G z_{k+1} - u_{k+1})
\]
Applying ADMM to More Than Two Functions

\[
\begin{align*}
\min_{z \in \mathbb{R}^d} \sum_{j=1}^{J} g_j(H^{(j)}z) & & \min_{z \in \mathbb{R}^d} f_2(Gz) \\
G &= \begin{bmatrix} H^{(1)} \\ \vdots \\ H^{(J)} \end{bmatrix} & u &= \begin{bmatrix} u^{(1)} \\ \vdots \\ u^{(J)} \end{bmatrix}
\end{align*}
\]

\[
\begin{align*}
z_{k+1} &= \left[ \sum_{j=1}^{J} (H^{(j)})^T H^{(j)} \right]^{-1} \left( \sum_{j=1}^{J} H^{(j)} \left( u_k^{(j)} + d_k^{(j)} \right) \right) \\
u_{k+1}^{(1)} &= \arg \min_u g_1(u) + \frac{\mu}{2} \| u - H^{(1)}z_{k+1} + d_k^{(1)} \|^2 = \text{prox}_{g_1/\mu} (H^{(1)}z_{k+1} - d_k^{(1)}) \\
&\vdots \\
u_{k+1}^{(J)} &= \arg \min_u g_J(u) + \frac{\mu}{2} \| u - H^{(J)}z_{k+1} + d_k^{(J)} \|^2 = \text{prox}_{g_J/\mu} (H^{(J)}z_{k+1} - d_k^{(J)}) \\
d_{k+1}^{(1)} &= d_k^{(1)} - (G^{(1)}z_{k+1} - u_{k+1}^{(1)}) \\
&\vdots \\
d_{k+1}^{(J)} &= d_k^{(J)} - (G^{(J)}z_{k+1} - u_{k+1}^{(J)})
\end{align*}
\]
Applying ADMM to More Than Two Functions

\[ z_{k+1} = \left( \sum_{j=1}^{J} (H^{(j)})^* H^{(j)} \right)^{-1} \left( \sum_{j=1}^{J} (H^{(j)})^* \left( u_k^{(j)} + d_k^{(j)} \right) \right) \]

\[
\begin{align*}
    u_{k+1}^{(1)} &= \arg \min_{u} g_1(u) + \frac{\mu}{2} \| u - H^{(1)} z_{k+1} + d_k^{(1)} \|^2 = \text{prox}_{g_1/\mu}(H^{(1)} z_{k+1} - d_k^{(1)}) \\
    \vdots & \quad \vdots \quad \vdots \\
    u_{k+1}^{(J)} &= \arg \min_{u} g_J(u) + \frac{\mu}{2} \| u - H^{(J)} z_{k+1} + d_k^{(J)} \|^2 = \text{prox}_{g_1/\mu}(H^{(1)} z_{k+1} - d_k^{(1)}) \\
    d_{k+1}^{(1)} &= d_k^{(1)} - (H^{(1)} z_{k+1} - u_k^{(1)}) \\
    \vdots & \quad \vdots \quad \vdots \\
    d_{k+1}^{(J)} &= d_k^{(J)} - (H^{(J)} z_{k+1} - u_k^{(J)})
\end{align*}
\]

Conditions for easy applicability: inexpensive matrix inversion

inexpensive proximity operators
Linear/Gaussian Observations: Frame-Based Analysis

Problem: \( \hat{x} \in \arg\min_x \frac{1}{2} \| A x - y \|_2^2 + \tau \| P x \|_1 \)

Template: \( \min_{z \in \mathbb{R}^d} \sum_{j=1}^J g_j(H^{(j)} z) \)

Mapping: \( J = 2, \quad g_1(z) = \frac{1}{2} \| z - y \|_2^2, \quad g_2(z) = \tau \| z \|_1 \)

\( H^{(1)} = A, \quad H^{(2)} = P, \)

Convergence conditions: \( g_1 \) and \( g_2 \) are closed, proper, and convex.

\( G = \begin{bmatrix} A \\ P \end{bmatrix} \) has full column rank.

Resulting algorithm: SALSA

(split augmented Lagrangian shrinkage algorithm) [Afonso, Bioucas-Dias and F., 09, 10]
Key steps of SALSA (both for analysis and synthesis):

Moreau proximity operator of 
\[ g_1(z) = \frac{1}{2} \| z - y \|_2^2, \]

\[
\text{prox}_{g_1/\mu}(u) = \arg \min_z \frac{1}{2 \mu} \| z - y \|_2^2 + \frac{1}{2} \| z - u \|_2^2 = \frac{y + \mu u}{1 + \mu}
\]

Moreau proximity operator of 
\[ g_2(z) = \tau \| z \|_1, \]

\[
\text{prox}_{g_2/\mu}(u) = \text{soft} \left( u, \frac{\tau}{\mu} \right)
\]

Linear step (next slide):

\[ z_{k+1} = \left( \sum_{j=1}^{J} (H^{(j)})^T H^{(j)} \right)^{-1} \left( \sum_{j=1}^{J} H^{(j)} \left( u_k^{(j)} + d_k^{(j)} \right) \right) \]
Handling the Matrix Inversion: Frame-Based Analysis

Frame-based analysis:
\[
\left( \sum_{j=1}^{J} (H^{(j)}*H^{(j)}) \right)^{-1} = \left( A*A + P*P \right)^{-1} = \left( A*A + I \right)^{-1}
\]

\[ P*P = I \]

Periodic deconvolution: \[ A = U*DU \]

Compressive imaging (MRI): \[ A = MU \]

Inpainting (recovery of lost pixels): \[ A = S \]

\[ O(n \log n) \]

\[ O(n) \]

\[ \frac{1}{2} U*M*MU \]

\[ S*S \] is diagonal

\[ S*S + I \] is a diagonal inversion
SALSA for Frame-Based Synthesis

Problem: \( \hat{x} \in \arg \min_x \frac{1}{2} \| B W x - y \|_2^2 + \tau \| x \|_1 \)

Template: \( \min \limits_{z \in \mathbb{R}^d} \sum_{j=1}^{J} g_j(H^{(j)}z) \)

Mapping: \( J = 2, \quad g_1(z) = \frac{1}{2} \| z - y \|_2^2, \quad g_2(z) = \tau \| z \|_1 \)

\( H^{(1)} = BW \quad \quad \quad H^{(2)} = I, \)

Convergence conditions: \( g_1 \) and \( g_2 \) are closed, proper, and convex.

\( G = \begin{bmatrix} B & W \\ I \end{bmatrix} \) has full column rank.
Handling the Matrix Inversion: Frame-Based Analysis

Frame-based analysis:
\[
\left[ \sum_{j=1}^{J} (H^{(j)})^* H^{(j)} \right]^{-1} = \left[ W^* B^* B W + I \right]^{-1}
\]

Periodic deconvolution:
\[
B = U^* D U
\]

Compressive imaging (MRI):
\[
B = M U
\]

Inpainting (recovery of lost pixels):
\[
B = S
\]

\[O(n \log n)\]

DFT

diagonal matrix

matrix inversion lemma + \(WW^* = I\)

subsample matrix: \(MM^* = I\)

subsample matrix: \(SS^* = I\)
SALSA Experiments

Benchmark problem: image deconvolution (9x9 uniform blur, 40dB BSNR)

undecimated Haar wavelets, $\ell_1$ synthesis regularization.
SALSA Experiments

Image inpainting (50% pixels missing)

<table>
<thead>
<tr>
<th>Alg.</th>
<th>Calls to $B, B^H$</th>
<th>Iter.</th>
<th>CPU time (sec.)</th>
<th>MSE</th>
<th>ISNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FISTA</td>
<td>1022</td>
<td>340</td>
<td>263.8</td>
<td>92.01</td>
<td>18.96</td>
</tr>
<tr>
<td>TwIST</td>
<td>271</td>
<td>124</td>
<td>112.7</td>
<td>100.92</td>
<td>18.54</td>
</tr>
<tr>
<td>SALSA</td>
<td>84</td>
<td>28</td>
<td>20.88</td>
<td>77.61</td>
<td>19.68</td>
</tr>
</tbody>
</table>
Frame-Based Analysis Deconvolution of Poissonian Images

Problem template: 
\[
\min_{\mathbf{u} \in \mathbb{R}^d} \sum_{j=1}^{J} g_j(\mathbf{H}(j) \mathbf{u}) \quad (P1)
\]

Frame-analysis regularization: 
\[
\hat{\mathbf{x}} \in \arg \min_{\mathbf{x}} \mathcal{L}_\mathbf{P}(\mathbf{B} \mathbf{x}) + \lambda \|\mathbf{P} \mathbf{x}\|_1 + \ell_{\mathbb{R}_+^n}(\mathbf{x})
\]

Same form as \((P1)\) with: 
\[ J = 3, \quad g_1 = \mathcal{L}_\mathbf{P}, \quad g_2 = \|\cdot\|_1, \quad g_3 = \ell_{\mathbb{R}_+^n} \]

Convergence conditions: \( g_1, g_2, \) and \( g_3 \) are closed, proper, and convex.

\[
\mathbf{G} = \begin{bmatrix} \mathbf{B} \\ \mathbf{P} \\ \mathbf{I} \end{bmatrix} \quad \text{has full column rank}
\]

Required inversion: 
\[
\left( \sum_{j=1}^{J} (\mathbf{H}(j))^* \mathbf{H}(j) \right)^{-1} = \left[ \mathbf{B}^* \mathbf{B} + \mathbf{P}^* \mathbf{P} + \mathbf{I} \right]^{-1} = \left[ \mathbf{B}^* \mathbf{B} + 2 \mathbf{I} \right]^{-1}
\]

...again, easy in deconvolution, inpainting, tomography.
Proximity Operator of the Poisson Log-Likelihood

Proximity operator of the Poisson log-likelihood

\[
\mathbf{u}_{k+1}^{(1)} \leftarrow \arg \min_{\mathbf{v}} \frac{\mu}{2} \left\| \mathbf{v} - \mathbf{v}_k^{(1)} \right\|_2^2 + \sum_{i=1}^{m} \xi(v_i, y_i)
\]

\[
\xi(z, y) = z + \nu_{\mathbb{R}_+}(z) - y \log(z_+)
\]

Separable problem with closed-form (non-negative) solution

\[
\mathbf{u}_{i,k+1}^{(1)} = \frac{1}{2} \left( \nu_{i,k}^{(1)} - \frac{1}{\mu} + \sqrt{\left( \nu_{i,k}^{(1)} - \frac{1}{\mu} \right)^2 + \frac{4y_i}{\mu}} \right)
\]

Proximity operator of \( g_3 = \nu_{\mathbb{R}_+^n} \) is simply \( \text{prox}_{\nu_{\mathbb{R}_+^n}}(\mathbf{x}) = (\mathbf{x})_+ \)
Poisson Image Deconvolution by AL (PIDAL): Experiments

Comparison with [Dupé, Fadili, Starck, 09] and [Starck, Bijaoui, Murtagh, 95]

<table>
<thead>
<tr>
<th>Image</th>
<th>$M$</th>
<th>PIDAL-TV</th>
<th>PIDAL-FA</th>
<th>[Dupé, Fadili, Starck, 09]</th>
<th>[Starck et al, 95]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>MAE</td>
<td>iterations</td>
<td>time</td>
<td>MAE</td>
</tr>
<tr>
<td>Cameraman</td>
<td>5</td>
<td>0.27</td>
<td>120</td>
<td>22</td>
<td>0.26</td>
</tr>
<tr>
<td>Cameraman</td>
<td>30</td>
<td>1.29</td>
<td>51</td>
<td>9.1</td>
<td>1.22</td>
</tr>
<tr>
<td>Cameraman</td>
<td>100</td>
<td>3.99</td>
<td>33</td>
<td>6.0</td>
<td>3.63</td>
</tr>
<tr>
<td>Cameraman</td>
<td>255</td>
<td>8.99</td>
<td>32</td>
<td>5.8</td>
<td>8.45</td>
</tr>
<tr>
<td>Neuron</td>
<td>5</td>
<td>0.17</td>
<td>117</td>
<td>3.6</td>
<td>0.18</td>
</tr>
<tr>
<td>Neuron</td>
<td>30</td>
<td>0.68</td>
<td>54</td>
<td>1.8</td>
<td>0.77</td>
</tr>
<tr>
<td>Neuron</td>
<td>100</td>
<td>1.75</td>
<td>43</td>
<td>1.4</td>
<td>2.04</td>
</tr>
<tr>
<td>Neuron</td>
<td>255</td>
<td>3.52</td>
<td>43</td>
<td>1.4</td>
<td>3.47</td>
</tr>
<tr>
<td>Cell</td>
<td>5</td>
<td>0.12</td>
<td>56</td>
<td>10</td>
<td>0.11</td>
</tr>
<tr>
<td>Cell</td>
<td>30</td>
<td>0.57</td>
<td>31</td>
<td>6.5</td>
<td>0.54</td>
</tr>
<tr>
<td>Cell</td>
<td>100</td>
<td>1.71</td>
<td>85</td>
<td>15</td>
<td>1.46</td>
</tr>
<tr>
<td>Cell</td>
<td>255</td>
<td>3.77</td>
<td>89</td>
<td>17</td>
<td>3.33</td>
</tr>
</tbody>
</table>

MAE $\equiv \frac{\|\hat{X} - X\|_1}{n}$
Constrained Optimization Formulation

Unconstrained optimization formulation: \[
\min_x \frac{1}{2} \|Ax - y\|_2^2 + \tau c(x)
\]

Constrained optimization formulation: \[
\min_x c(x) \\
\text{s.t. } \|Ax - y\|_2^2 \leq \varepsilon
\]

basis pursuit denoising (BPN) [Chen, Donoho, Saunders, 1998]

Both analysis and synthesis can be used:

- frame-based analysis, \[ c(x) = \|Px\|_1 \]
- frame-based synthesis \[ c(x) = \|x\|_1 \]

\[
A = BW
\]
Proposed Approach for Constrained Formulation

Constrained problem: \[
\min_x c(x) \\
\text{s.t. } \|Ax - y\|_2^2 \leq \varepsilon
\]

...can be written as \[
\min_x c(x) + \lambda_{E(\varepsilon,y)}(Ax)
\]

\[E(\varepsilon, y) = \{x \in \mathbb{R}^n : \|x - y\|_2 \leq \varepsilon\} \quad \lambda_S(z) = \begin{cases} 0 & \iff z \in S \\ +\infty & \iff z \notin S \end{cases}
\]

...which has the form \[
\min_{u \in \mathbb{R}^d} \sum_{j=1}^{J} g_j(H^{(j)}u) \quad (P1)
\]

with \[J = 2, \quad g_1(z) = c(z), \quad H^{(1)} = I \quad G = \begin{bmatrix} I \\ A \end{bmatrix}
\]

\[g_2(z) = \lambda_{E(\varepsilon,y)}(z), \quad H^{(2)} = A \quad \text{full column rank}
\]

Resulting algorithm: C-SALSA (constrained-SALSA) [Afonso, Bioucas-Dias, F, 09, 11]
Some Aspects of C-SALSA

Moreau proximity operator of $\ell_E(\varepsilon, y)$ is simply a projection on an $\ell_2$ ball:

$$\text{prox}_{\ell_E(\varepsilon, y)}(u) = \arg \min_z \ell_E(\varepsilon, y) + \frac{1}{2} \| z - u \|_2^2$$

$$= \begin{cases} u & \iff \| u - y \|_2 \leq \varepsilon \\ y + \frac{\varepsilon(u - y)}{\|u - y\|_2} & \iff \| u - y \|_2 > \varepsilon \end{cases}$$

As SALSA, also C-SALSA involves inversion of the form

$$\left[ W^* B^* B W + I \right]^{-1} \quad \text{or} \quad \left[ A^* A + P^* P \right]^{-1}$$

...all the same tricks as above.
C-SALSA Experiments: Image Deblurring

Benchmark experiments:

<table>
<thead>
<tr>
<th>Experiment</th>
<th>blur kernel</th>
<th>$\sigma^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$9 \times 9$ uniform</td>
<td>0.56^2</td>
</tr>
<tr>
<td>2A</td>
<td>Gaussian</td>
<td>2</td>
</tr>
<tr>
<td>2B</td>
<td>Gaussian</td>
<td>8</td>
</tr>
<tr>
<td>3A</td>
<td>$h_{ij} = 1/(1 + i^2 + j^2)$</td>
<td>2</td>
</tr>
<tr>
<td>3B</td>
<td>$h_{ij} = 1/(1 + i^2 + j^2)$</td>
<td>8</td>
</tr>
</tbody>
</table>

Comparison with

NESTA [Bobin, Becker, Candès, 09]

SPGL1 [van den Berg, Friedlander, 08]

Frame-synthesis

<table>
<thead>
<tr>
<th>Expt.</th>
<th>Avg. calls to $B$, $B^H$ (min/max)</th>
<th>Iterations</th>
<th>CPU time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SPGL1, NESTA, C-SALSA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1029 (659/1290)</td>
<td>3520 (3501/3541)</td>
<td>398 (388/406)</td>
</tr>
<tr>
<td>2A</td>
<td>511 (279/663)</td>
<td>4897 (4777/4981)</td>
<td>451 (442/460)</td>
</tr>
<tr>
<td>2B</td>
<td>377 (141/532)</td>
<td>3397 (3345/3473)</td>
<td>362 (355/370)</td>
</tr>
<tr>
<td>3A</td>
<td>675 (378/772)</td>
<td>2622 (2589/2661)</td>
<td>172 (166/175)</td>
</tr>
<tr>
<td>3B</td>
<td>404 (300/475)</td>
<td>2446 (2401/2485)</td>
<td>134 (130/136)</td>
</tr>
</tbody>
</table>

Frame-analysis

<table>
<thead>
<tr>
<th>Expt.</th>
<th>Avg. calls to $B$, $B^H$ (min/max)</th>
<th>Iterations</th>
<th>CPU time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SPGL1, NESTA, C-SALSA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2881 (2861/2889)</td>
<td>413 (404/419)</td>
<td>720</td>
</tr>
<tr>
<td>2A</td>
<td>2451 (2377/2505)</td>
<td>362 (344/371)</td>
<td>613</td>
</tr>
<tr>
<td>2B</td>
<td>2139 (2065/2197)</td>
<td>290 (278/299)</td>
<td>535</td>
</tr>
<tr>
<td>3A</td>
<td>2203 (2181/2217)</td>
<td>137 (134/143)</td>
<td>551</td>
</tr>
</tbody>
</table>

TV

<table>
<thead>
<tr>
<th>Expt.</th>
<th>Avg. calls to $B$, $B^H$ (min/max)</th>
<th>Iterations</th>
<th>CPU time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SPGL1, NESTA, C-SALSA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>7783 (7767/7795)</td>
<td>695 (680/710)</td>
<td>1945</td>
</tr>
<tr>
<td>2A</td>
<td>7323 (7291/7351)</td>
<td>559 (536/578)</td>
<td>1830</td>
</tr>
<tr>
<td>2B</td>
<td>6828 (6775/6883)</td>
<td>299 (269/329)</td>
<td>1707</td>
</tr>
<tr>
<td>3A</td>
<td>6594 (6513/6661)</td>
<td>176 (98/209)</td>
<td>1649</td>
</tr>
<tr>
<td>3B</td>
<td>5514 (5417/5585)</td>
<td>108 (104/110)</td>
<td>1379</td>
</tr>
</tbody>
</table>
(Overlapping) Group Regularization (Structured Sparsity)

\[
\hat{x} \in \arg \min_{x \in \mathbb{R}^n} \frac{1}{2} \|Ax - y\|_2^2 + \sum_{i=1}^k \lambda_i \phi_i(x_{G_i})
\]

If groups have hierarchical structure and the \(\phi_i\) are \(\ell_1, \ell_2,\) or \(\ell_\infty\) norms, then \(\text{prox} \sum_i \phi_i\)
can be computed efficiently \[\text{Jenatton, Audibert, Bach, 2009}\]

Algorithm for arbitrary groups with \(\phi_i = \| \cdot \|_2\) (FoGLASSO) \[\text{Liu and Ye, 2010}\]

ADMM allows addressing this problem, if…
\[\text{F and Bioucas-Dias, SPARS'2011}\]
…the functions \(\phi_i\) have simple \(\text{prox} \phi_i\)

…a certain matrix inversion can be efficiently handled

See \[\text{Qin and Goldfarb, 2011}\] for another ADM method for this problem.
(Overlapping) Group Regularization

\[
\hat{x} \in \arg \min_{x \in \mathbb{R}^n} \frac{1}{2} \|Ax - y\|_2^2 + \sum_{i=1}^{k} \lambda_i \phi_i(x_{G_i})
\]

Template: \[
\min_u \sum_{j=1}^{J} g_j(H^{(j)} u)
\]

Mapping: \[J = k + 1,\]

\[
g_1(z) = \frac{1}{2} \|z - y\|_2^2,
\]

\[
g_j(z) = \lambda_j \phi_j(z), \quad j = 2, \ldots, k + 1
\]

\[H^{(1)} = A,\]

\[H^{(j)} = \text{diag}(I_{1\in G_j}, I_{2\in G_j}, \ldots, I_{n\in G_j}), \quad j = 2, \ldots, k + 1\]
(Overlapping) Group Regularization: Toy Example

\[ n = 200, \ A \in \mathbb{R}^{100 \times 200} \quad \text{(i.i.d. } \mathcal{N}(0, 1)) \quad y = Ax + n, \ n \sim \mathcal{N}(0, 10^{-2}) \]

\[ \phi_i = \| \cdot \|_2, \ k = 19, \ G_1 = \{1, \ldots, 20\}, \ G_2 = \{11, \ldots, 30\}, \ldots \]
Hybrid: Analysis + Synthesis Regularization

\[ \hat{x} \in \arg \min_x \frac{1}{2} \|B W x - y\|^2_2 + \tau_1 \|x\|_1 + \tau_2 \|P W x\|_1 \]

Observation matrix ↑ synthesis matrix ↑ analysis matrix of
of a Parseval frame another Parseval frame

As in frame-based “synthesis” regularization,

\[ x \]
contains representation coefficients (not the signal itself)

these coefficients are under regularization

As in frame-based “analysis” regularization,

\[ z = W x \]
is “analyzed”:
\[ P z \]
the result of the analysis is under regularization
Hybrid: Analysis + Synthesis Regularization

Problem: \[ \hat{x} \in \arg \min_{x} \frac{1}{2} \| B W x - y \|_2^2 + \tau_1 \| x \|_1 + \tau_2 \| P W x \|_1 \]

Template: \[ \min_{u \in \mathbb{R}^d} \sum_{j=1}^{J} g_j(H^{(j)} u) \quad (P1) \]

Mapping: \( J = 3 \), \( g_1(z) = \frac{1}{2} \| z - y \|_2^2 \), \( g_2(z) = \tau_1 \| z \|_1 \), \( g_3(z) = \tau_2 \| z \|_1 \)

\[ H^{(1)} = B W, \quad H^{(2)} = I, \quad H^{(3)} = P W, \]

Convergence conditions: all \( g_i \) are closed, proper, and convex.

\[ G = \begin{bmatrix} B W & I & P W \end{bmatrix} \] has full column rank.
Experiments: Image Deconvolution

Benchmark experiments:

<table>
<thead>
<tr>
<th>Experiment</th>
<th>blur kernel</th>
<th>$\sigma^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$9 \times 9$ uniform</td>
<td>0.56$^2$</td>
</tr>
<tr>
<td>2A</td>
<td>Gaussian</td>
<td>2</td>
</tr>
<tr>
<td>2B</td>
<td>Gaussian</td>
<td>8</td>
</tr>
<tr>
<td>3A</td>
<td>$h_{ij} = 1/(1 + i^2 + j^2)$</td>
<td>2</td>
</tr>
<tr>
<td>3B</td>
<td>$h_{ij} = 1/(1 + i^2 + j^2)$</td>
<td>8</td>
</tr>
</tbody>
</table>

Two different frames (undecimated Daubechies 2 and 6); hand-tuned parameters.

<table>
<thead>
<tr>
<th>Exp.</th>
<th>Analysis ISNR</th>
<th>Synthesis ISNR</th>
<th>Hybrid ISNR</th>
<th>Analysis time (sec)</th>
<th>Synthesis time (sec)</th>
<th>Hybrid time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8.52 dB</td>
<td>7.13 dB</td>
<td>8.61 dB</td>
<td>34.1</td>
<td>4.1</td>
<td>11.1</td>
</tr>
<tr>
<td>2A</td>
<td>5.38 dB</td>
<td>4.49 dB</td>
<td>5.48 dB</td>
<td>21.3</td>
<td>1.4</td>
<td>3.8</td>
</tr>
<tr>
<td>2B</td>
<td>5.27 dB</td>
<td>4.48 dB</td>
<td>5.39 dB</td>
<td>20.2</td>
<td>1.6</td>
<td>3.4</td>
</tr>
<tr>
<td>3A</td>
<td>7.33 dB</td>
<td>6.32 dB</td>
<td>7.46 dB</td>
<td>17.9</td>
<td>1.6</td>
<td>3.7</td>
</tr>
<tr>
<td>3B</td>
<td>4.93 dB</td>
<td>4.37 dB</td>
<td>5.31 dB</td>
<td>20.1</td>
<td>2.5</td>
<td>3.9</td>
</tr>
</tbody>
</table>

Preliminary conclusions: analysis is better than synthesis
hybrid is slightly better than analysis
hybrid is faster than analysis
Yet Another Application: Spectral Unmixing

Goal: find the relative abundance of each “material” in each pixel.

\[ \hat{x} \in \arg\min_x \frac{1}{2} \| A x - y \|_2^2 + \nu_{\mathbb{R}_+^n}(x) + \nu_{\{1\}}(1^T x) \]

Given library of spectra

Indicator of the canonical simplex
Spectral Unmixing

Problem: \( \hat{x} \in \arg \min_x \frac{1}{2} \|Ax - y\|_2^2 + \nu_{\mathbb{R}_+^n}(x) + \nu_{\{1\}}(1^T x) \)

Template: \( \min_{u \in \mathbb{R}^d} \sum_{j=1}^J g_j(H^{(j)} u) \)

Mapping: \( J = 3, \ g_1(z) = \frac{1}{2} \|z - y\|_2^2, \ g_2(z) = \nu_{\mathbb{R}_+^n}(z) \), \( g_3(z) = \nu_{\{1\}}(z) \)

\( H^{(1)} = A, \ H^{(2)} = I, \ H^{(3)} = 1^T \)

Proximity operators are trivial.

Matrix inversion can be precomputed (typical sizes 200~300 \( \times \) 500~1000)

\[
\left[ \sum_{j=1}^J (H^{(j)})^* H^{(j)} \right]^{-1} = \left[ A^T A + I + 11^T \right]^{-1}
\]

Spectral unmixing by split augmented Lagrangian (SUNSAL) [Bioucas-Dias, F, 2010]

Related algorithm (split-Bregman view) in [Szlam, Guo, Osher, 2010]
Summary, Open Questions, and Ongoing Work

**Summary:** ADMM is a very flexible and efficient tool, for a variety of optimization problems arising in imaging inverse problems…

…if a certain matrix can be cheaply inverted.

**Ongoing work:** efficiently handling (large) problems where the matrix inversion can’t be sidestepped (L-BFGS [Afonso, Bioucas-Dias, F, 2010])
(alternating linearization [Goldfarb, Ma, Scheinberg, 2010],
primal-dual methods [Chambolle, Pock, 2009], [Esser, Zhang, Chan, 2009])

hyperspectral imaging with spatial regularization

(overlapping) group regularization
[F, Bioucas-Dias, SPARS’11]

MAP inference in graphical models (dual decomposition + ADMM)
[Martins, Smith, Xing, Aguiar, F, ICML’2011]

logistic regression, …
Some Publications


