Covariant LEAst-square Re-fitting for image restoration

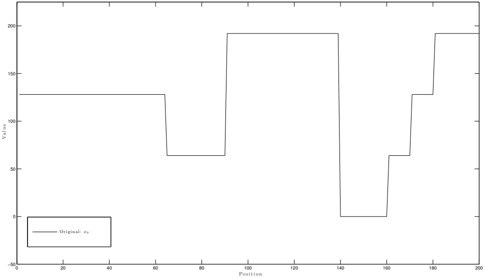
N. Papadakis[1]
Joint work with C.-A. Deledalle[2], J. Salmon[3] & S. Vaiter[4]

[1]CNRS, Institut de Mathématiques de Bordeaux
[2] University California San Diego
[3] Université de Montpellier, IMAG
[4] CNRS, Institut de Mathématiques de Bourgogne

Variational methods and optimization in imaging

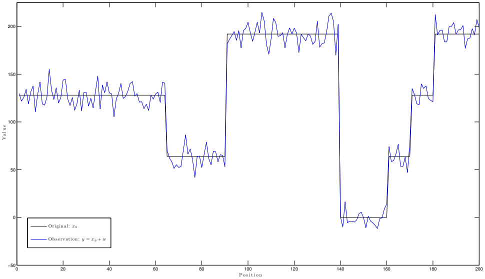
February 5th 2019

What is Re-fitting?



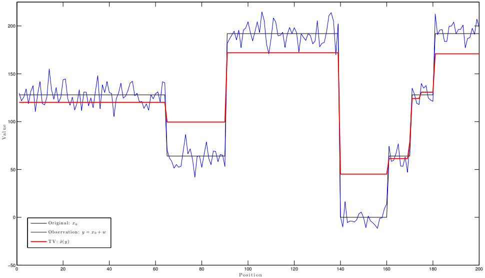
True piecewise constant signal x_0

What is Re-fitting?



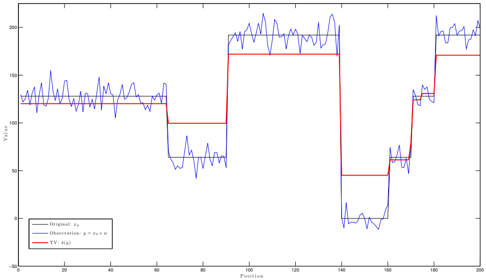
Observations $y = x_0 + w$

What is Re-fitting?



Total Variation (TV) restoration $\hat{x}(y) = \underset{x}{\operatorname{argmin}} \frac{1}{2}||y-x||^2 + \lambda||\nabla x||_1$

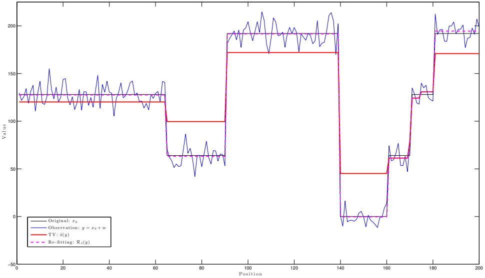
What is Re-fitting?



Total Variation (TV) restoration $\hat{x}(y) = \underset{x}{\operatorname{argmin}} \frac{1}{2}||y - x||^2 + \lambda||\nabla x||_1$

- Problem: $\hat{x}(y)$ is biased [Strong and Chan 1996], i.e., $\mathbb{E}\hat{x}(y) \neq x_0$.

What is Re-fitting?



Re-fitting $\mathcal{R}_{\hat{x}}(y)$

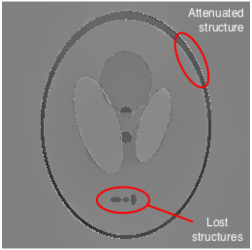
- Problem: $\hat{x}(y)$ is biased [Strong and Chan 1996], i.e., $\mathbb{E}\hat{x}(y) \neq x_0$.
- A solution: Compute $\mathcal{R}_{\hat{x}}(y)$ as the mean of y on each piece of $\hat{x}(y)$

Which bias reduction with Re-fitting?

Noise free image

Noisy data

Anisotropic TV



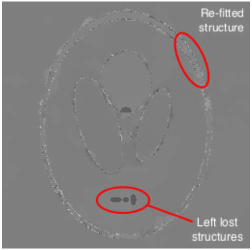
Error

Which bias reduction with Re-fitting?

Noise free image

Noisy data

Re-fitting



Error

Re-fitting arbitrary models

Can we generalize this approach to other estimators than TV?

- General setting

$$\hat{x}(y) = \underset{x}{\operatorname{argmin}}\, F(x, y) + \lambda G(x)$$

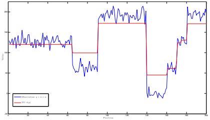
- Data fidelity w.r.t observations y: F convex
- Prior model on the solution: G convex
- Regularization parameter: $\lambda > 0$

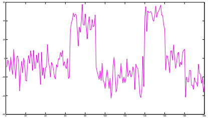
For inverse problems

- $F(x, y) = F(\Phi x - y)$
- Φ is a linear operator: convolution, mask...

Related works

Twicing [Tukey, 1977], Bregman iterations [Osher et al. 2005, 2015, Burger et al. 2006], Boosting [Elad et al. 2007, 2015, Milanfar 2013]...

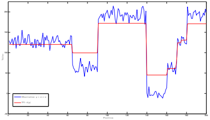


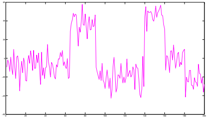


Apply the model to the residual $y - \Phi\hat{x}(y)$

Related works

Twicing [Tukey, 1977], Bregman iterations [Osher et al. 2005, 2015, Burger et al. 2006], Boosting [Elad et al. 2007, 2015, Milanfar 2013]...





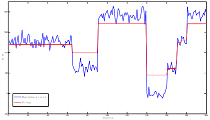
Apply the model to the residual $y - \Phi\hat{x}(y)$

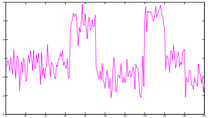
Iterative process

- $x_0 = \underset{x}{\operatorname{argmin}}\ F(\Phi x - y) + \lambda G(x)$
- $x_k = x_{k-1} + \underset{x}{\operatorname{argmin}}\ F(\Phi x - (y - \Phi x_{k-1})) + \lambda_k G(x)$

Related works

Twicing [Tukey, 1977], Bregman iterations [Osher et al. 2005, 2015, Burger et al. 2006], Boosting [Elad et al. 2007, 2015, Milanfar 2013]...





Apply the model to the residual $y - \Phi\hat{x}(y)$

Iterative process

- $x_0 = \underset{x}{\operatorname{argmin}}\ F(\Phi x - y) + \lambda G(x)$
- $x_k = x_{k-1} + \underset{x}{\operatorname{argmin}}\ F(\Phi x - (y - \Phi x_{k-1})) + \lambda_k G(x)$

Limitations

- Support not preserved
- How many iterations?
- Varying parameters λ_k

Objectives

Automatic re-fitting process

$$\hat{x}(y) = \underset{x}{\operatorname{argmin}}\, F(x, y) + \lambda G(x)$$

- Keep structures and regularity present in the biased solution
- Correct the model bias
- No additional parameter

Objectives

Automatic re-fitting process

$$\hat{x}(y) = \underset{x}{\operatorname{argmin}}\, F(x, y) + \lambda G(x)$$

- Keep structures and regularity present in the biased solution
- Correct the model bias
- No additional parameter

Handle existing "black box" algorithms $\hat{x}(y)$

- Non-Local Means [Buades et al. 2005]
- BM3D [Dabov et al. 2007], DDID [Knaus & Zwicker 2013]
- Trained Deep Networks

Objectives

Automatic re-fitting process

$$\hat{x}(y) = \underset{x}{\operatorname{argmin}}\, F(x, y) + \lambda G(x)$$

- Keep structures and regularity present in the biased solution
- Correct the model bias
- No additional parameter

Handle existing "black box" algorithms $\hat{x}(y)$

- Non-Local Means [Buades et al. 2005]
- BM3D [Dabov et al. 2007], DDID [Knaus & Zwicker 2013]
- Trained Deep Networks

Bias-variance trade-off

- Bias reduction is not always favorable in terms of MSE
- Re-fitting re-injects part of the variance

Objectives

Automatic re-fitting process

$$\hat{x}(y) = \operatorname*{argmin}_x F(x, y) + \lambda G(x)$$

- Keep structures and regularity present in the biased solution
- Correct the model bias
- No additional parameter

Handle existing "black box" algorithms $\hat{x}(y)$

- Non-Local Means [Buades et al. 2005]
- BM3D [Dabov et al. 2007], DDID [Knaus & Zwicker 2013]
- Trained Deep Networks

Bias-variance trade-off

- Bias reduction is not always favorable in terms of MSE
- Re-fitting re-injects part of the variance

MSE is not expected to be reduced with re-fitting

Formalizing Re-fitting for TV

- Linear inverse problem:

$$y = \Phi x_0 + w, \quad \underbrace{y \in \mathbb{R}^p}_{\text{observation}}, \quad \underbrace{x_0 \in \mathbb{R}^n}_{\text{signal of interest}}, \underbrace{\Phi \in \mathbb{R}^{p \times n}}_{\text{linear operator}}, \quad \underbrace{\mathbb{E}[w] = 0_p}_{\text{white noise}}$$

- TV regularization:

$$\hat{x}(y) = \underset{x}{\operatorname{argmin}} \frac{1}{2}||\Phi x - y||^2 + \lambda||\nabla x||_1$$

- Re-fitting TV (constrained least-square [Efron et al. 2004, Lederer 2013])

$$\tilde{x}(y) \in \underset{x \in \mathcal{M}_{\hat{x}}(y)}{\operatorname{argmin}} \|\Phi x - y\|^2$$

  with $\mathcal{M}_{\hat{x}}(y)$ the model subspace:

$$\mathcal{M}_{\hat{x}}(y) = \{x \in \mathbb{R}^n \setminus \forall i, (\nabla \hat{x}(y))_i = 0 \Rightarrow (\nabla x)_i = 0\}$$

- Set of signals with same jumps (co-support)

Generalizing TV Re-fitting procedure

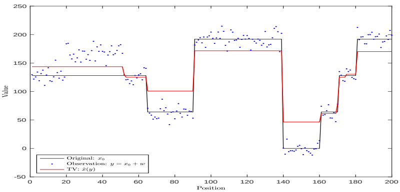
- Reformulating without the notion of jumps
- Understanding what is captured by $\mathcal{M}_{\hat{x}}(y)$

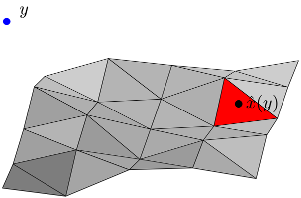
Idea: $\mathcal{M}_{\hat{x}}(y)$ captures linear invariances of $\hat{x}(y)$ w.r.t small perturbations on y

Invariant Re-fitting [Deledalle, P. & Salmon 2015]

- Piece-wise affine mapping $y \mapsto \hat{x}(y)$

$$\hat{x}(y) = \underset{x}{\operatorname{argmin}}\, F(\Phi x, y) + \lambda G(x)$$

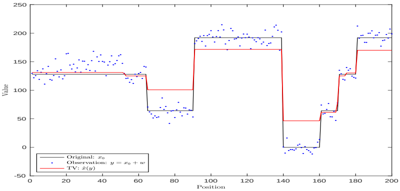


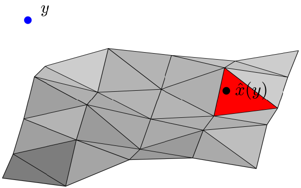


Invariant Re-fitting [Deledalle, P. & Salmon 2015]

- Piece-wise affine mapping $y \mapsto \hat{x}(y)$

$$\hat{x}(y) = \underset{x}{\operatorname{argmin}}\ F(\Phi x, y) + \lambda G(x)$$

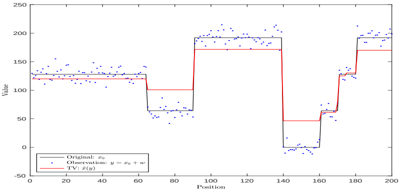


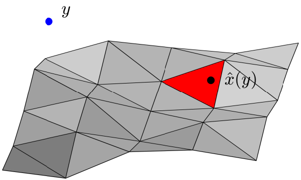


Invariant Re-fitting [Deledalle, P. & Salmon 2015]

- Piece-wise affine mapping $y \mapsto \hat{x}(y)$

$$\hat{x}(y) = \underset{x}{\operatorname{argmin}}\ F(\Phi x, y) + \lambda G(x)$$





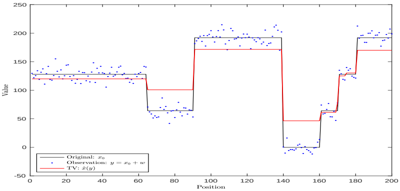
Invariant Re-fitting [Deledalle, P. & Salmon 2015]

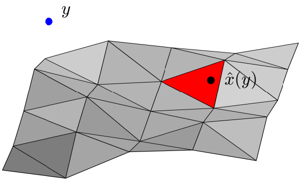
- Piece-wise affine mapping $y \mapsto \hat{x}(y)$

$$\hat{x}(y) = \operatorname*{argmin}_x F(\Phi x, y) + \lambda G(x)$$

- Jacobian of the estimator:

$$(J_{\hat{x}(y)})_{ij} = \frac{\partial \hat{x}(y)_i}{\partial y_j}$$

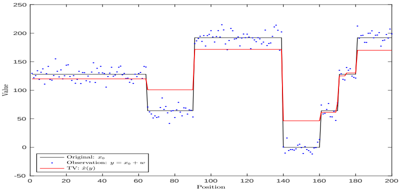




Invariant Re-fitting [Deledalle, P. & Salmon 2015]

- Piece-wise affine mapping $y \mapsto \hat{x}(y)$

$$\hat{x}(y) = \underset{x}{\operatorname{argmin}}\, F(\Phi x, y) + \lambda G(x)$$

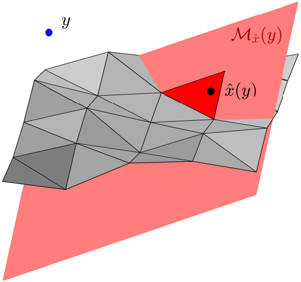


- Jacobian of the estimator:

$$(J_{\hat{x}(y)})_{ij} = \frac{\partial \hat{x}(y)_i}{\partial y_j}$$

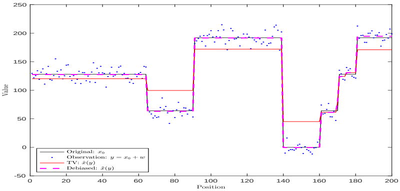
- Model subspace:

$$\mathcal{M}_{\hat{x}}(y) = \hat{x}(y) + \mathrm{Im}[J_{\hat{x}(y)}]$$



Tangent space of the mapping

Invariant Re-fitting [Deledalle, P. & Salmon 2015]



- Piece-wise affine mapping $y \mapsto \hat{x}(y)$

$$\hat{x}(y) = \underset{x}{\operatorname{argmin}}\ F(\Phi x, y) + \lambda G(x)$$

- Jacobian of the estimator:

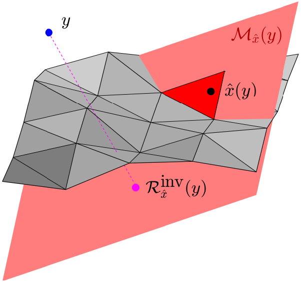
$$(J_{\hat{x}(y)})_{ij} = \frac{\partial \hat{x}(y)_i}{\partial y_j}$$

- Model subspace:

$$\color{red}{\mathcal{M}_{\hat{x}}(y)} = \hat{x}(y) + \operatorname{Im}[J_{\hat{x}(y)}]$$

- Invariant Least square Re-fitting:

$$\mathcal{R}^{\text{inv}}_{\hat{x}}(y) = \underset{x \in \color{red}{\mathcal{M}_{\hat{x}}(y)}}{\operatorname{argmin}}\ \|\Phi x - y\|^2$$



Tangent space of the mapping

Practical Re-fitting for ℓ_1 analysis minimization

$$\hat{x}(y) = \operatorname*{argmin}_x \frac{1}{2}||\Phi x - y||^2 + \lambda||\Gamma x||_1 \tag{1}$$

Remark: $\Gamma = \mathrm{Id}$ is the LASSO, $\Gamma = [\nabla_x, \nabla_y]^\top$ is the Anisotropic TV.

Numerical stability issue

- Piecewise constant solution [Strong & Chan 2003, Caselles et al., 2009]
- Re-fitting $\hat{x}(y)$ requires the support: $\hat{\mathcal{I}} = \{i \in [1, m], \text{ s.t. } |\Gamma\hat{x}|_i > 0\}$
- But in practice, $\hat{x}$ is only approximated through a converging sequence $\hat{x}^k$
- Unfortunately, $\hat{x}^k \approx \hat{x} \not\Rightarrow \hat{\mathcal{I}}^k \approx \hat{\mathcal{I}}$
- Illustration for Anisotropic TV denoising ($\Phi = \mathrm{Id}$):


Blurry obs. y


Biased $\hat{x}$


estimate $\hat{x}^k$


Re-fitting $\hat{x}$


Re-fitting $\hat{x}^k$

Practical Re-fitting for ℓ_1 analysis minimization

$$\hat{x}(y) = \underset{x}{\operatorname{argmin}} \ \frac{1}{2}||\Phi x - y||^2 + \lambda||\Gamma x||_1 \tag{1}$$

Remark: $\Gamma = \mathrm{Id}$ is the LASSO, $\Gamma = [\nabla_x, \nabla_y]^\top$ is the Anisotropic TV.

Proposed approach

- Provided $\operatorname{Ker}\Phi \cap \operatorname{Ker}\Gamma = \{0\}$, one has for (1) [Vaiter et al. 2016]:

$$\mathcal{R}^{\mathrm{inv}}_{\hat{x}}(y) = J_y[y] \quad \text{with} \quad J_y = \left.\frac{\partial \hat{x}(y)}{\partial y}\right|_y$$

- Interpretation: $\mathcal{R}^{\mathrm{inv}}_{\hat{x}}(y)$ is the derivative of $\hat{x}(y)$ in the direction of y
- Algorithm: Compute $\tilde{x}^k$ by chain rule as the derivative of $\hat{x}^k(y)$ in the direction of y
- Question: Does $\tilde{x}^k$ converge towards $\mathcal{R}^{\mathrm{inv}}_{\hat{x}}(y)$?

Practical Re-fitting for ℓ_1 analysis minimization

$$\hat{x}(y) = \underset{x}{\operatorname{argmin}} \frac{1}{2}||\Phi x - y||^2 + \lambda||\Gamma x||_1 \tag{1}$$

Remark: $\Gamma = \mathrm{Id}$ is the LASSO, $\Gamma = [\nabla_x, \nabla_y]^\top$ is the Anisotropic TV.

Proposed approach

- Provided $\mathrm{Ker}\,\Phi \cap \mathrm{Ker}\,\Gamma = \{0\}$, one has for (1) [Vaiter et al. 2016]:

$$\mathcal{R}_{\hat{x}}^{\mathrm{inv}}(y) = J_y[y] \quad \text{with} \quad J_y = \left.\frac{\partial \hat{x}(y)}{\partial y}\right|_y$$

- Interpretation: $\mathcal{R}_{\hat{x}}^{\mathrm{inv}}(y)$ is the derivative of $\hat{x}(y)$ in the direction of y
- Algorithm: Compute $\tilde{x}^k$ by chain rule as the derivative of $\hat{x}^k(y)$ in the direction of y
- Question: Does $\tilde{x}^k$ converge towards $\mathcal{R}_{\hat{x}}^{\mathrm{inv}}(y)$?

yes, at least for the ADMM or the Chambolle-Pock sequences

Implementation for Anisotropic TV

$$\hat{x}(y) = \operatorname*{argmin}_x \frac{1}{2}||y - x||^2 + \lambda||\Gamma x||_1$$

- Primal-dual implementation [Chambolle and Pock 2011]:

$$\begin{cases} z^{k+1} = \mathrm{Proj}_{B_\lambda}\left(z^k + \sigma\Gamma x^k\right) \\ x^{k+1} = \frac{x^k + \tau\left(y - \Gamma^\top\left(z^{k+1}\right)\right)}{1+\tau} \end{cases}$$

- Projection: $\mathrm{Proj}_{B_\lambda}(z)_i = \begin{cases} z_i & \text{if } |z_i| \leqslant \lambda \\ \lambda \mathrm{sign}(z_i) & \text{otherwise} \end{cases}$

Implementation for Anisotropic TV

$$\hat{x}(y) = \operatorname*{argmin}_x \frac{1}{2}||y-x||^2 + \lambda||\Gamma x||_1$$

- Primal-dual implementation [Chambolle and Pock 2011]:

$$\left\{\begin{array}{lcl} z^{k+1} & = & \mathrm{Proj}_{B_\lambda}\left(z^k + \sigma\Gamma x^k\right) \\ \tilde{z}^{k+1} & = & P_{z^k+\sigma\Gamma x^k}\left(\tilde{z}^k + \sigma\Gamma\tilde{x}^k\right) \\ x^{k+1} & = & \frac{x^k+\tau\left(y-\Gamma^\top\left(z^{k+1}\right)\right)}{1+\tau} \\ \tilde{x}^{k+1} & = & \frac{\tilde{x}^k+\tau\left(y-\Gamma^\top\left(\tilde{z}^{k+1}\right)\right)}{1+\tau} \end{array}\right.$$

- Projection: $$\mathrm{Proj}_{B_\lambda}(z)_i = \left\{\begin{array}{ll} z_i & \text{if } |z_i| \leqslant \lambda \\ \lambda\mathrm{sign}(z_i) & \text{otherwise} \end{array}\right.$$

$$P_z = \left\{\begin{array}{ll} \mathrm{Id} & \text{if } |z_i| \leqslant \lambda + \beta \\ 0 & \text{otherwise} \end{array}\right.$$

Implementation for Anisotropic TV

$$\hat{x}(y) = \underset{x}{\operatorname{argmin}} \frac{1}{2}||y - x||^2 + \lambda||\Gamma x||_1$$

- Primal-dual implementation [Chambolle and Pock 2011]:

$$\begin{cases} z^{k+1} & = & \mathrm{Proj}_{B_\lambda}\left(z^k + \sigma\Gamma x^k\right) \\ \tilde{z}^{k+1} & = & P_{z^k+\sigma\Gamma x^k}\left(\tilde{z}^k + \sigma\Gamma\tilde{x}^k\right) \\ x^{k+1} & = & \frac{x^k+\tau\left(y-\Gamma^\top\left(z^{k+1}\right)\right)}{1+\tau} \\ \tilde{x}^{k+1} & = & \frac{\tilde{x}^k+\tau\left(y-\Gamma^\top\left(\tilde{z}^{k+1}\right)\right)}{1+\tau} \end{cases}$$

- Projection: $$\mathrm{Proj}_{B_\lambda}(z)_i = \begin{cases} z_i & \text{if } |z_i| \leqslant \lambda \\ \lambda\mathrm{sign}(z_i) & \text{otherwise} \end{cases}$$

$$P_z = \begin{cases} \mathrm{Id} & \text{if } |z_i| \leqslant \lambda + \beta \\ 0 & \text{otherwise} \end{cases}$$

Theorem: *The sequence $\tilde{x}^k$ converges to the re-fitting $\mathcal{R}^{inv}_{\hat{x}}(y)$ of $\hat{x}(y)$, $\forall\beta > 0$ s.t. $\beta < \sigma|\Gamma\hat{x}(y)|_i$, $\forall i \in \mathcal{I}$*

Implementation for Anisotropic TV

$$\hat{x}(y) = \underset{x}{\operatorname{argmin}} \frac{1}{2}||y - x||^2 + \lambda||\Gamma x||_1$$

- Primal-dual implementation [Chambolle and Pock 2011]:

$$\begin{cases} z^{k+1} & = & \mathrm{Proj}_{B_\lambda}\left(z^k + \sigma\Gamma x^k\right) \\ \tilde{z}^{k+1} & = & P_{z^k+\sigma\Gamma x^k}\left(\tilde{z}^k + \sigma\Gamma\tilde{x}^k\right) \\ x^{k+1} & = & \frac{x^k+\tau\left(y-\Gamma^\top\left(z^{k+1}\right)\right)}{1+\tau} \\ \tilde{x}^{k+1} & = & \frac{\tilde{x}^k+\tau\left(y-\Gamma^\top\left(\tilde{z}^{k+1}\right)\right)}{1+\tau} \end{cases}$$

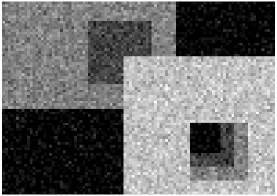
- Projection: $\mathrm{Proj}_{B_\lambda}(z)_i = \begin{cases} z_i & \text{if } |z_i| \leqslant \lambda \\ \lambda\mathrm{sign}(z_i) & \text{otherwise} \end{cases}$

$$P_z = \begin{cases} \mathrm{Id} & \text{if } |z_i| \leqslant \lambda + \beta \\ 0 & \text{otherwise} \end{cases}$$

Theorem: *The sequence $\tilde{x}^k$ converges to the re-fitting $\mathcal{R}^{inv}_{\hat{x}}(y)$ of $\hat{x}(y)$, $\forall\beta > 0$ s.t. $\beta < \sigma|\Gamma\hat{x}(y)|_i$, $\forall i \in \mathcal{I}$*

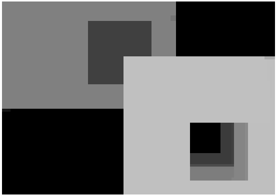
Complexity: twice that of the Chambolle-Pock algorithm.

Anisotropic TV: illustration



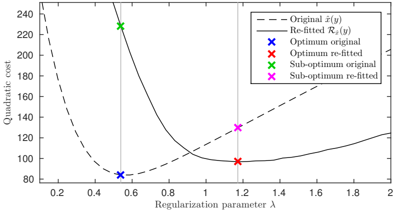
y

$\hat{x}(y)$



$\mathcal{R}^{\text{inv}}_{\hat{x}}(y)$

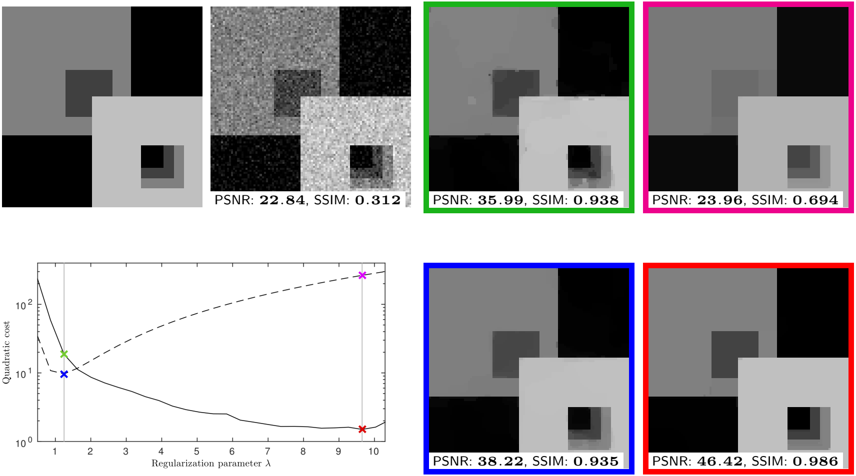
Anisotropic TV: Bias-variance trade-off



Anisotropic TV

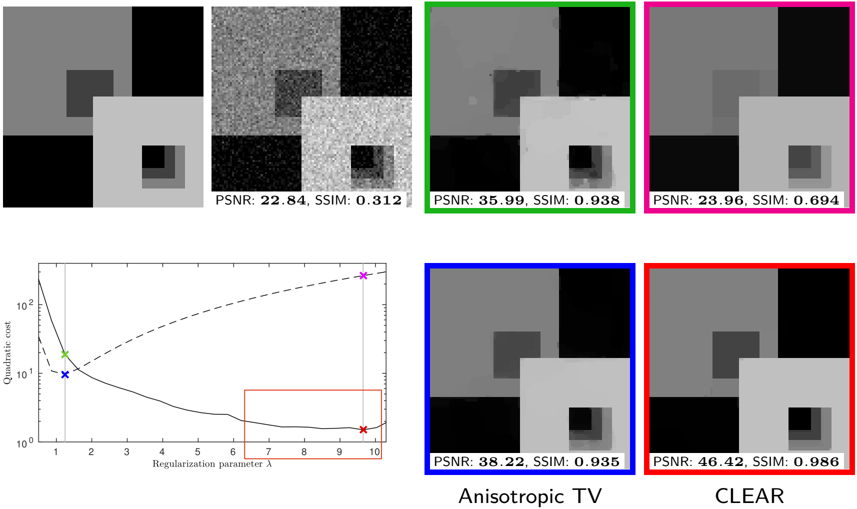
CLEAR

Anisotropic TV: Bias-variance trade-off

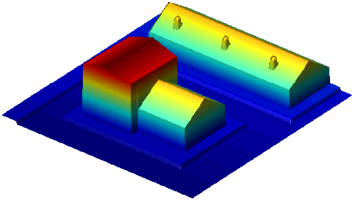


Anisotropic TV CLEAR

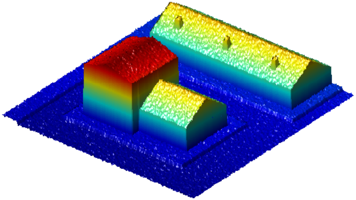
Anisotropic TV: Bias-variance trade-off



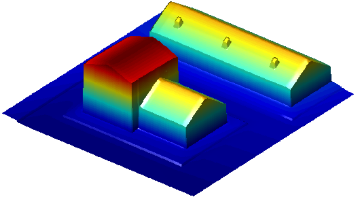
Example: Anisotropic TGV



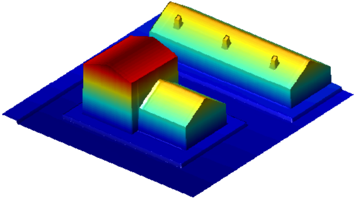
x_0



y



$\hat{x}(y)$



$R^{\mathrm{inv}}_{\hat{x}}(y)$

Example: Isotropic TV

- Restoration model for image y

$$\hat{x}(y) = \underset{x}{\operatorname{argmin}} \frac{1}{2}||y - x||^2 + \lambda||\nabla x||_{1,2}$$

- Model subspace of isotropic TV is the same than anisotropic TV:

signals whose gradients share their support with $\nabla \hat{x}(y)$

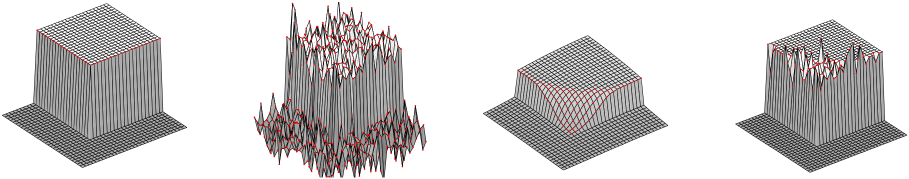
Example: Isotropic TV

- Restoration model for image y

$$\hat{x}(y) = \underset{x}{\operatorname{argmin}} \frac{1}{2}||y - x||^2 + \lambda||\nabla x||_{1,2}$$

- Model subspace of isotropic TV is the same than anisotropic TV:

signals whose gradients share their support with $\nabla \hat{x}(y)$



Noise free | Noisy data y | $\hat{x}(y)$ | $\mathcal{R}^{\text{inv}}_{\hat{x}}(y)$

Non sparse support: noise is re-injected

Illustration done with an ugly standard (i.e. non Condat and non Chambolle-Pock) discretization of isotropic TV

Limitations

Model subspace

- Only captures linear invariances w.r.t. small perturbations of y

Jacobian matrix

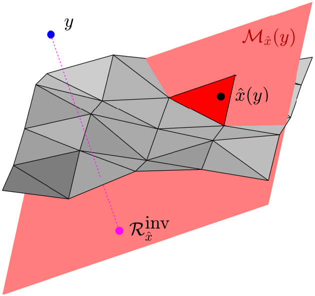
- Captures desirable covariant relationships between the entries of y and the entries of $\hat{x}(y)$ that should be preserved [Deledalle, P., Salmon and Vaiter, 2017, 2019]

Least-square Re-fitting

General problem

$$\hat{x}(y) = \underset{x}{\operatorname{argmin}}\ F(\Phi x, y) + \lambda G(x)$$

- Φ linear operator, F and G convex



Least-square Re-fitting

General problem

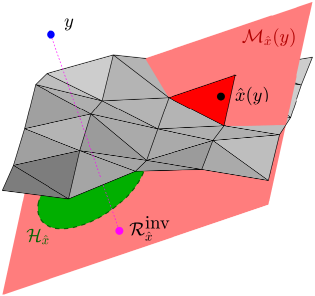
$$\hat{x}(y) = \underset{x}{\operatorname{argmin}}\ F(\Phi x, y) + \lambda G(x)$$

- Φ linear operator, F and G convex

Desirable properties of Re-fitting operator

$h \in \mathcal{H}_{\hat{x}}$ iff

1. $h \in \mathcal{M}_{\hat{x}}(y)$
2. Affine map: $h(y) = Ay + b$
3. Preservation of covariants: $J_h(y) = \rho J_{\hat{x}}(y)$
4. Coherence: $h(\Phi \hat{x}(y)) = \hat{x}(y)$



Least-square Re-fitting

General problem

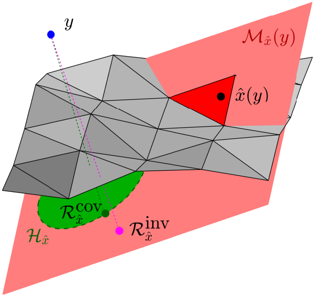
$$\hat{x}(y) = \underset{x}{\operatorname{argmin}}\; F(\Phi x, y) + \lambda G(x)$$

- Φ linear operator, F and G convex

Desirable properties of Re-fitting operator

$h \in \mathcal{H}_{\hat{x}}$ iff

1. $h \in \mathcal{M}_{\hat{x}}(y)$
2. Affine map: $h(y) = Ay + b$
3. Preservation of covariants: $J_h(y) = \rho J_{\hat{x}}(y)$
4. Coherence: $h(\Phi\hat{x}(y)) = \hat{x}(y)$



Covariant LEAst-square Re-fitting

$$\mathcal{R}^{\text{cov}}_{\hat{x}}(y) = \underset{x \in \mathcal{H}_{\hat{x}}}{\operatorname{argmin}}\; \frac{1}{2}||\Phi x - y||^2$$

Covariant LEAst-square Re-fitting

Proposition

The covariant Re-fitting has an explicit formulation

$$\mathcal{R}^{cov}_{\hat{x}}(y) = \hat{x}(y) + \rho J(y - \Phi\hat{x}(y)) = \underset{x \in \mathcal{H}_{\hat{x}}}{\operatorname{argmin}} \frac{1}{2}||\Phi x - y||^2$$

where for $\delta = y - \Phi\hat{x}(y)$*:*

$$\rho = \begin{cases} \frac{\langle \Phi J\delta, \delta \rangle}{||\Phi J\delta||^2} & \textit{if } \Phi J\delta \neq 0 \\ 1 & \textit{otherwise} \end{cases}$$

Covariant LEAst-square Re-fitting

Proposition

The covariant Re-fitting has an explicit formulation

$$\mathcal{R}_{\hat{x}}^{cov}(y) = \hat{x}(y) + \rho J(y - \Phi\hat{x}(y)) = \underset{x \in \mathcal{H}_{\hat{x}}}{\operatorname{argmin}} \frac{1}{2}||\Phi x - y||^2$$

where for $\delta = y - \Phi\hat{x}(y)$*:*

$$\rho = \begin{cases} \frac{\langle \Phi J\delta, \delta\rangle}{||\Phi J\delta||^2} & \textit{if } \Phi J\delta \neq 0 \\ 1 & \textit{otherwise} \end{cases}$$

Properties

- If ΦJ is an orthogonal projector, $\rho = 1$ and $\Phi\mathcal{R}_{\hat{x}}^{\mathrm{cov}}(y) = \Phi\mathcal{R}_{\hat{x}}^{\mathrm{inv}}(y)$

Covariant LEAst-square Re-fitting

Proposition

The covariant Re-fitting has an explicit formulation

$$\mathcal{R}_{\hat{x}}^{cov}(y) = \hat{x}(y) + \rho J(y - \Phi\hat{x}(y)) = \underset{x\in\mathcal{H}_{\hat{x}}}{\operatorname{argmin}} \frac{1}{2}||\Phi x - y||^2$$

where for $\delta = y - \Phi\hat{x}(y)$*:*

$$\rho = \begin{cases} \frac{\langle \Phi J\delta, \delta\rangle}{||\Phi J\delta||^2} & \text{if } \Phi J\delta \neq 0 \\ 1 & \text{otherwise} \end{cases}$$

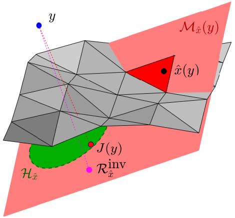
Properties

- If ΦJ is an orthogonal projector, $\rho = 1$ and $\Phi\mathcal{R}_{\hat{x}}^{\mathrm{cov}}(y) = \Phi\mathcal{R}_{\hat{x}}^{\mathrm{inv}}(y)$
- If F convex, G convex and 1-homogenous and

$$\hat{x}(y) = \underset{x}{\operatorname{argmin}}\, F(\Phi x - y) + G(x),$$

then $J\Phi\hat{x}(y) = \hat{x}(y)$ a.e. so that:

$$\mathcal{R}_{\hat{x}}^{\mathrm{cov}}(y) = (1-\rho)\hat{x}(y) + \rho J y$$



Covariant LEAst-square Re-fitting

Proposition

The covariant Re-fitting has an explicit formulation

$$\mathcal{R}^{cov}_{\hat{x}}(y) = \hat{x}(y) + \rho J(y - \Phi\hat{x}(y)) = \underset{x \in \mathcal{H}_{\hat{x}}}{\operatorname{argmin}} \frac{1}{2}||\Phi x - y||^2$$

where for $\delta = y - \Phi\hat{x}(y)$*:*

$$\rho = \begin{cases} \frac{\langle \Phi J\delta, \delta\rangle}{||\Phi J \delta||^2} & \text{if } \Phi J\delta \neq 0 \\ 1 & \text{otherwise} \end{cases}$$

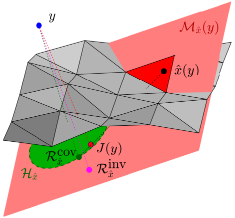
Properties

- If ΦJ is an orthogonal projector, $\rho = 1$ and $\Phi\mathcal{R}^{\text{cov}}_{\hat{x}}(y) = \Phi\mathcal{R}^{\text{inv}}_{\hat{x}}(y)$
- If F convex, G convex and 1-homogenous and

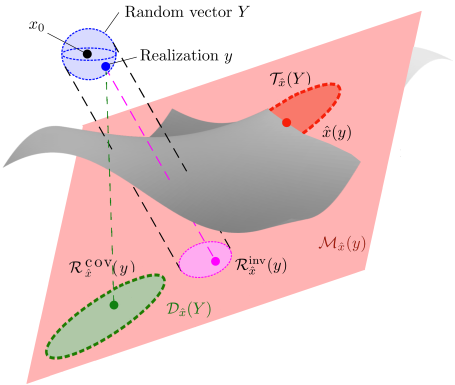
$$\hat{x}(y) = \underset{x}{\operatorname{argmin}}\, F(\Phi x - y) + G(x),$$

then $J\Phi\hat{x}(y) = \hat{x}(y)$ a.e. so that:

$$\mathcal{R}^{\text{cov}}_{\hat{x}}(y) = (1-\rho)\hat{x}(y) + \rho J y$$



Statistical interpretation

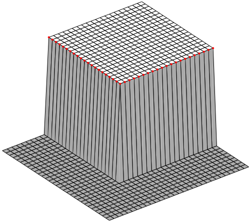


Theorem (Bias reduction)

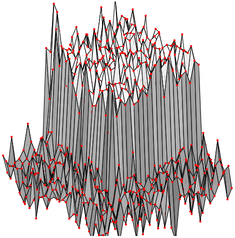
If ΦJ is an orthogonal projector or ρ satisfies technical conditions

$$||\Phi(\mathbb{E}[\mathcal{D}_{\hat{x}}(Y)] - x_0)||_2 \leqslant ||\Phi(\mathbb{E}[\mathcal{T}_{\hat{x}}(Y)] - x_0)||_2$$

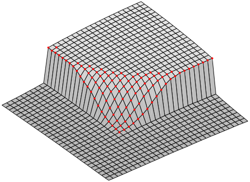
Example: Isotropic TV



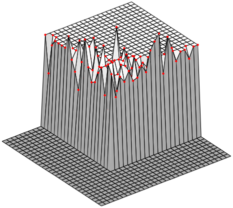
Noise free



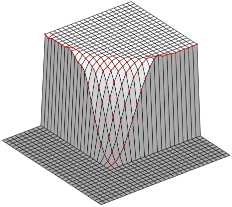
Noisy data y



$\hat{x}(y)$



$\mathcal{R}^{\text{inv}}_{\hat{x}}(y)$

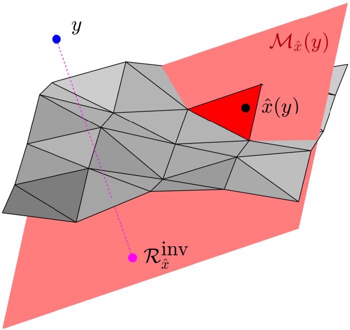


$\mathcal{R}^{\text{cov}}_{\hat{x}}(y)$

Why not iterating as Boosting approaches?

- Differentiable estimator w.r.t y:

$$\tilde{x}_0 = \hat{x}(y) = \underset{x}{\operatorname{argmin}} \, \|x-y\|^2 + \lambda \|\nabla x\|_{1,2}$$



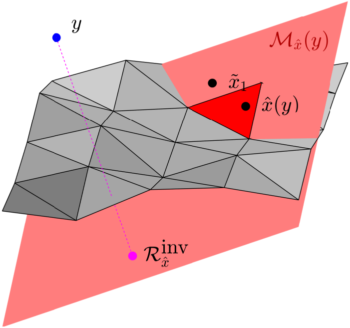
Why not iterating as Boosting approaches?

- Differentiable estimator w.r.t y:

$$\tilde{x}_0 = \hat{x}(y) = \underset{x}{\text{argmin}} \; \|x - y\|^2 + \lambda \|\nabla x\|_{1,2}$$

- **Bregman iterations**:

$$\tilde{x}_{k+1} = \underset{x}{\text{argmin}} \; \|x - y\|^2 + \lambda D_{\|\nabla . \|_{1,2}}(x, \tilde{x}_k)$$



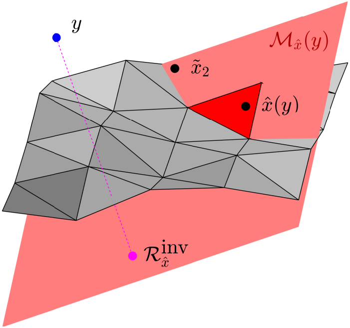
Why not iterating as Boosting approaches?

- Differentiable estimator w.r.t y:

$$\tilde{x}_0 = \hat{x}(y) = \underset{x}{\operatorname{argmin}} \; \|x - y\|^2 + \lambda \|\nabla x\|_{1,2}$$

- **Bregman iterations**:

$$\tilde{x}_{k+1} = \underset{x}{\operatorname{argmin}} \; \|x - y\|^2 + \lambda D_{\|\nabla \cdot\|_{1,2}}(x, \tilde{x}_k)$$



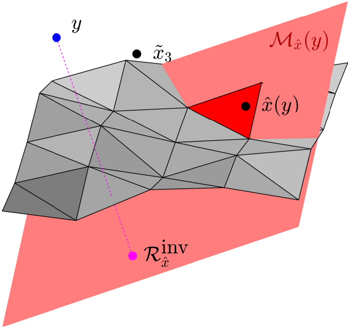
Why not iterating as Boosting approaches?

- Differentiable estimator w.r.t y:

$$\tilde{x}_0 = \hat{x}(y) = \underset{x}{\operatorname{argmin}} \; \|x - y\|^2 + \lambda \|\nabla x\|_{1,2}$$

- **Bregman iterations**:

$$\tilde{x}_{k+1} = \underset{x}{\operatorname{argmin}} \; \|x - y\|^2 + \lambda D_{\|\nabla \cdot\|_{1,2}}(x, \tilde{x}_k)$$



Why not iterating as Boosting approaches?

- Differentiable estimator w.r.t y:

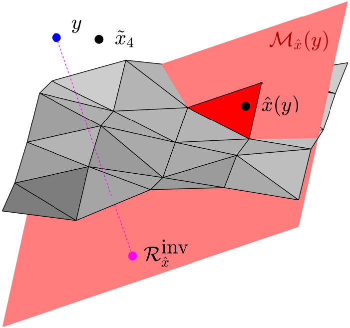
$$\tilde{x}_0 = \hat{x}(y) = \underset{x}{\operatorname{argmin}} \|x - y\|^2 + \lambda \|\nabla x\|_{1,2}$$

- **Bregman iterations**:

$$\tilde{x}_{k+1} = \underset{x}{\operatorname{argmin}} \|x - y\|^2 + \lambda D_{\|\nabla . \|_{1,2}}(x, \tilde{x}_k)$$

- Convergence:

$$\tilde{x}_k \to y$$



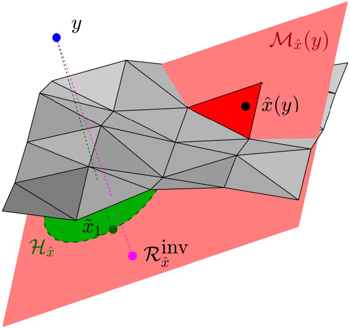
Why not iterating as Boosting approaches?

- Differentiable estimator w.r.t y:

$$\tilde{x}_0 = \hat{x}(y) = \underset{x}{\operatorname{argmin}} \|x - y\|^2 + \lambda \|\nabla x\|_{1,2}$$

- **Covariant iterations**:

$$\tilde{x}^{k+1}(y) = \tilde{x}^k(y) + \rho J(y - \Phi \tilde{x}^k(y))$$



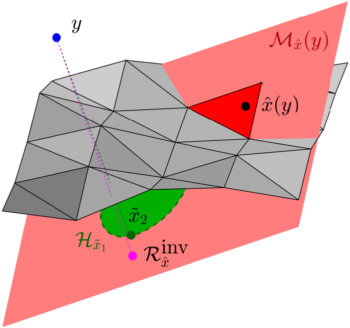
Why not iterating as Boosting approaches?

- Differentiable estimator w.r.t y:

$$\tilde{x}_0 = \hat{x}(y) = \underset{x}{\operatorname{argmin}} \; ||x - y||^2 + \lambda ||\nabla x||_{1,2}$$

- **Covariant iterations**:

$$\tilde{x}^{k+1}(y) = \tilde{x}^k(y) + \rho J(y - \Phi \tilde{x}^k(y))$$



Why not iterating as Boosting approaches?

- Differentiable estimator w.r.t y:

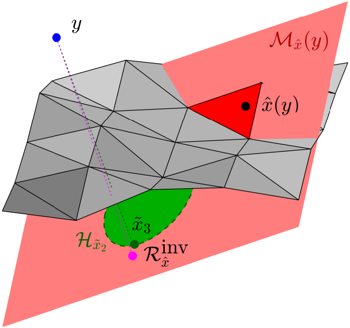
$$\tilde{x}_0 = \hat{x}(y) = \underset{x}{\operatorname{argmin}}\ \|x-y\|^2+\lambda\|\nabla x\|_{1,2}$$

- **Covariant iterations**:

$$\tilde{x}^{k+1}(y) = \tilde{x}^k(y) + \rho J(y - \Phi\tilde{x}^k(y))$$

- Convergence:

$$\tilde{x}^k(y) \to \mathcal{R}^{\text{inv}}_{\hat{x}}(y)$$



How computing the covariant Re-fitting?

- Explicit expression:

$$\mathcal{R}^{\text{cov}}_{\hat{x}}(y) = \hat{x}(y) + \rho J\delta$$

with $J = \frac{\partial \hat{x}(y)}{\partial y}$, $\delta = y - \Phi\hat{x}(y)$ and

$$\rho = \begin{cases} \frac{\langle \Phi J\delta, \delta\rangle}{||\Phi J\delta||^2} & \text{if } \Phi J\delta \neq 0 \\ 1 & \text{otherwise} \end{cases}$$

Main issue

- Being able to compute $J\delta$

Application of the Jacobian matrix to a vector

Algorithmic differentiation

- Iterative algorithm to obtain $\hat{x}(y)$:

$$x^{k+1} = \psi(x^k, y)$$

- Differentiation in the direction δ:

$$\begin{cases} x^{k+1} & = \psi(x^k, y) \\ \tilde{x}^{k+1} & = \partial_1 \psi(x^k, y)\tilde{x}^k + \partial_2 \psi(x^k, y)\delta \end{cases}$$

- $J_{x^k}(y)\delta = \tilde{x}^k$
- Joint estimation of x^k and $J_{x^k}(y)\delta$
- Double the computational cost

Application of the Jacobian matrix to a vector

Finite difference based differentiation

- $\hat{x}(y)$ can be any black box algorithm
- Directional derivative w.r.t to direction δ:

$$J_{\hat{x}}(y)\delta \approx \frac{\hat{x}(y+\varepsilon\delta)-\hat{x}(y)}{\varepsilon}$$

- Need to apply twice the algorithm

Computation of the Re-fitting

Covariant LEAst-square Re-fitting

$$\mathcal{R}_{\hat{x}}^{\text{cov}}(y) = \hat{x}(y) + \rho J\delta, \qquad \text{with } \delta = y - \Phi\hat{x}(y) \text{ and } \rho = \frac{\langle J\delta, \delta\rangle}{||J\delta||_2^2}$$

Two-steps with any denoising algorithm

1. Apply algorithm to y to get $\hat{x}(y)$ and set $\delta = y - \Phi\hat{x}(y)$
2. Compute $J\delta$ (with algorithmic or finite difference based differentiation)

Computation of the Re-fitting

Covariant LEAst-square Re-fitting

$$\mathcal{R}_{\hat{x}}^{\text{cov}}(y) = \hat{x}(y) + \rho J\delta, \qquad \text{with } \delta = y - \Phi\hat{x}(y) \text{ and } \rho = \frac{\langle J\delta, \delta\rangle}{||J\delta||_2^2}$$

Two-steps with any denoising algorithm

1. Apply algorithm to y to get $\hat{x}(y)$ and set $\delta = y - \Phi\hat{x}(y)$
2. Compute $J\delta$ (with algorithmic or finite difference based differentiation)

One-step for absolutely 1-homogeneous regularizer

Re-fitting simplifies to

$$\mathcal{R}_{\hat{x}}^{\text{cov}}(y) = (1-\rho)\hat{x}(y) + \rho J y$$

1. Estimate jointly $\hat{x}(y)$ and Jy with the differentiated algorithm

1 step Implementation: Anisotropic TV

- Model with 1-homogeneous regularizer:

$$\hat{x}(y) = \underset{x}{\operatorname{argmin}} \frac{1}{2}||y - x||^2 + \lambda||\nabla x||_1$$

- Primal-dual implementation [Chambolle and Pock 2011]:

$$\begin{cases} z^{k+1} & = & \mathrm{Proj}_{B_\lambda}\left(z^k + \sigma\nabla x^k\right) \\ \tilde{z}^{k+1} & = & P_{z^k+\sigma\nabla x^k}\left(\tilde{z}^k + \sigma\nabla\tilde{x}^k\right) \\ x^{k+1} & = & \frac{x^k+\tau\left(y+\mathrm{div}\left(z^{k+1}\right)\right)}{1+\tau} \\ \tilde{x}^{k+1} & = & \frac{\tilde{x}^k+\tau\left(y+\mathrm{div}\left(\tilde{z}^{k+1}\right)\right)}{1+\tau} \end{cases}$$

- Projection:

$$\mathrm{Proj}_{B_\lambda}(z)_i = \begin{cases} z_i & \text{if } |z_i| \leqslant \lambda \\ \lambda \mathrm{sign}(z_i) & \text{otherwise} \end{cases}$$

$$P_z = \begin{cases} \mathrm{Id} & \text{if } |z_i| \leqslant \lambda + \beta \\ 0 & \text{otherwise} \end{cases}$$

- $x^k \to \hat{x}(y)$ and $\tilde{x}^k = J_{x^k} y \to J_{\hat{x}} y$
- J is an orthogonal projector: $\mathcal{R}^{\mathrm{cov}}_{\hat{x}}(y) = \mathcal{R}^{\mathrm{inv}}_{\hat{x}}(y) = J_{\hat{x}} y$

1 step Implementation: Isotropic TV

- Model with 1-homogeneous regularizer:

$$\hat{x}(y) = \underset{x}{\text{argmin}} \ \frac{1}{2}||y-x||^2 + \lambda||\nabla x||_{1,2}$$

- Primal-dual implementation [Chambolle and Pock 2011]:

$$\begin{cases} z^{k+1} & = & \mathrm{Proj}_{B_\lambda}\left(z^k + \sigma \nabla x^k\right) \\ x^{k+1} & = & \frac{x^k + \tau\left(y + \mathrm{div}\left(z^{k+1}\right)\right)}{1+\tau} \end{cases}$$

- Projection:

$$\mathrm{Proj}_{B_\lambda}(z)_i = \begin{cases} z_i & \text{if } ||z_i||_2 \leqslant \lambda \\ \lambda \frac{z_i}{||z_i||_2} & \text{otherwise} \end{cases}$$

1 step Implementation: Isotropic TV

- Model with 1-homogeneous regularizer:

$$\hat{x}(y) = \underset{x}{\operatorname{argmin}} \frac{1}{2}||y - x||^2 + \lambda||\nabla x||_{1,2}$$

- Primal-dual implementation [Chambolle and Pock 2011]:

$$\begin{cases} z^{k+1} & = & \operatorname{Proj}_{B_\lambda}\left(z^k + \sigma \nabla x^k\right) \\ \tilde{z}^{k+1} & = & P_{z^k + \sigma \nabla x^k}\left(\tilde{z}^k + \sigma \nabla \tilde{x}^k\right) \\ x^{k+1} & = & \frac{x^k + \tau\left(y + \operatorname{div}\left(z^{k+1}\right)\right)}{1+\tau} \\ \tilde{x}^{k+1} & = & \frac{\tilde{x}^k + \tau\left(y + \operatorname{div}\left(\tilde{z}^{k+1}\right)\right)}{1+\tau} \end{cases}$$

- Projection:

$$\operatorname{Proj}_{B_\lambda}(z)_i = \begin{cases} z_i & \text{if } ||z_i||_2 \leqslant \lambda \\ \lambda \frac{z_i}{||z_i||_2} & \text{otherwise} \end{cases}$$

$$P_z = \begin{cases} \mathrm{Id} & \text{if } \|z_i\| \leqslant \lambda + \beta \\ \frac{\lambda}{||z_i||_2}\left(\mathrm{Id} - \frac{z_i z_i^\top}{||z_i||_2^2}\right) & \text{otherwise} \end{cases}$$

- $x^k \rightarrow \hat{x}(y)$ and $\tilde{x}^k = J_{x^k} y \rightarrow \tilde{x}$

1 step Implementation: Isotropic TV

- Model with 1-homogeneous regularizer:

$$\hat{x}(y) = \underset{x}{\operatorname{argmin}}\ \frac{1}{2}||y-x||^2 + \lambda||\nabla x||_{1,2}$$

- Primal-dual implementation [Chambolle and Pock 2011]:

$$\begin{cases} z^{k+1} &= \operatorname{Proj}_{B_\lambda}\left(z^k + \sigma\nabla x^k\right) \\ \tilde{z}^{k+1} &= P_{z^k+\sigma\nabla x^k}\left(\tilde{z}^k + \sigma\nabla\tilde{x}^k\right) \\ x^{k+1} &= \frac{x^k+\tau\left(y+\operatorname{div}\left(z^{k+1}\right)\right)}{1+\tau} \\ \tilde{x}^{k+1} &= \frac{\tilde{x}^k+\tau\left(y+\operatorname{div}\left(\tilde{z}^{k+1}\right)\right)}{1+\tau} \end{cases}$$

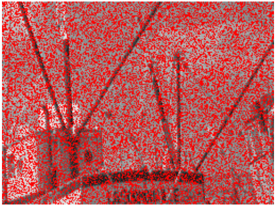
- Projection:

$$\operatorname{Proj}_{B_\lambda}(z)_i = \begin{cases} z_i & \text{if } ||z_i||_2 \leqslant \lambda \\ \lambda\frac{z_i}{||z_i||_2} & \text{otherwise} \end{cases}$$

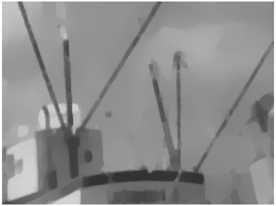
$$P_z = \begin{cases} \mathrm{Id} & \text{if } \|z_i\| \leqslant \lambda+\beta \\ \frac{\lambda}{||z_i||_2}\left(\mathrm{Id} - \frac{z_i z_i^\top}{||z_i||_2^2}\right) & \text{otherwise} \end{cases}$$

- $x^k \to \hat{x}(y)$ and $\tilde{x}^k = J_{x^k}y \to \tilde{x}$
- Covariant re-fitting: $\mathcal{R}_{\hat{x}}^{\text{cov}}(y) = (1-\rho)\hat{x} + \rho\tilde{x}$, with $\rho = \frac{\langle\tilde{x}-\hat{x}, y-\hat{x}\rangle}{||\tilde{x}-\hat{x}||_2^2}$

Inpainting with Isotropic TV



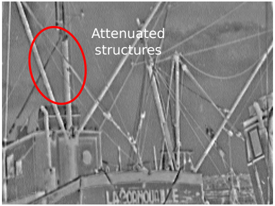
y



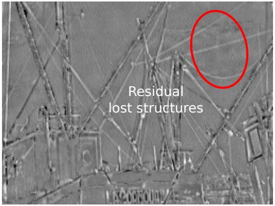
$\hat{x}(y)$

$R_{\hat{x}}^{\text{cov}}(y)$

x_0

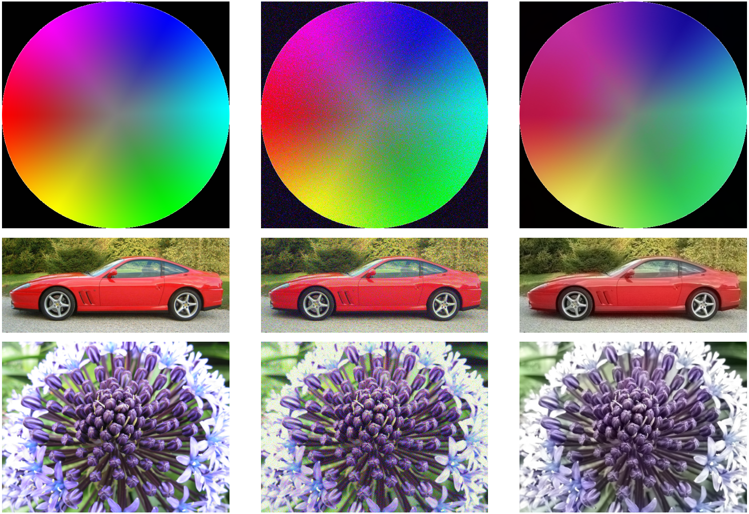


$||\hat{x}(y) - x_0||_2$



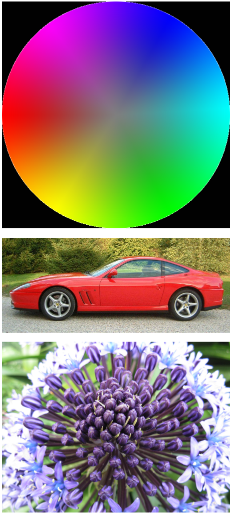
$||R_{\hat{x}}^{\text{cov}}(y) - x_0||_2$

Extension to chrominance [Pierre, Aujol, Deledalle, P., 2017]

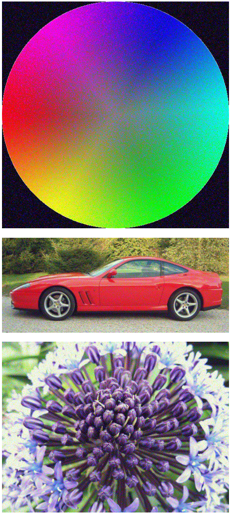


Noise free image | Noisy image | Denoised image

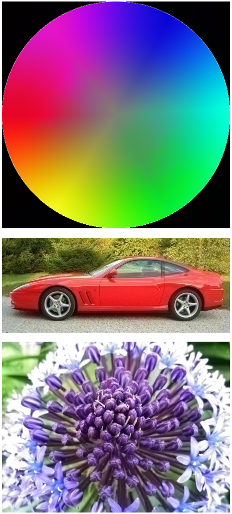
Extension to chrominance [Pierre, Aujol, Deledalle, P., 2017]



Noise free image



Noisy image



Re-fitting

2 steps Implementation: Non-Local Means

- Model without 1-homogeneous regularizer:

$$\hat{x}(y)_i = \frac{\sum_j w_{ij}^y y_j}{\sum_j w_{ij}^y}, \quad w_{i,j}^y = \exp\left(-\|\mathcal{P}_i y - \mathcal{P}_j y\|_2^2 / h^2\right)$$

- Differentiate NLM code
- Algorithm:
- Re-fitting with algorithmic differentiation:
  1: Run NLM code $\hat{x}(y)$ and set $\delta = y - \hat{x}(y)$
  2: Run differentiated NLM code in the direction δ to get $J\delta$
  3: Set $\rho = \frac{\langle J\delta, \delta \rangle}{||J\delta||_2^2}$
  4: Covariant re-fitting: $\mathcal{R}_{\hat{x}}^{\text{cov}}(y) = \hat{x}(y) + \rho J\delta$

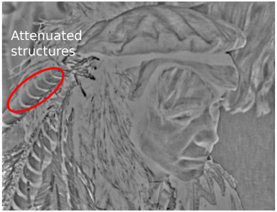
Non-Local Means

y

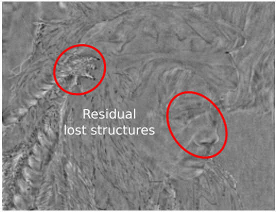
$\hat{x}(y)$

$R_{\hat{x}}^{\text{cov}}(y)$

x_0

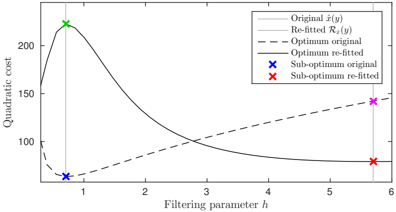


$||\hat{x}(y) - x_0||_2$



$||R_{\hat{x}}^{\text{cov}}(y) - x_0||_2$

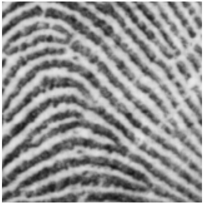
Non-Local Means: Bias-variance trade-off

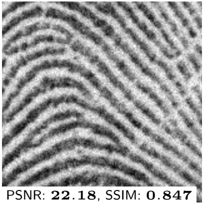


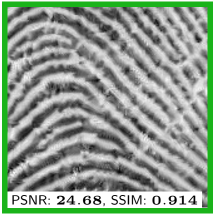
Non-Local Means

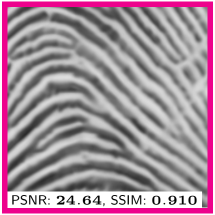
CLEAR

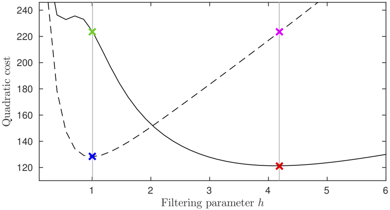
Bias-variance trade-off: Non-Local Means

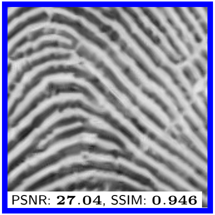




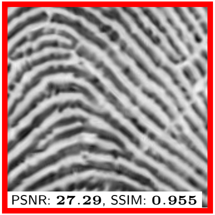






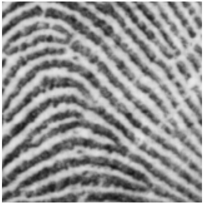


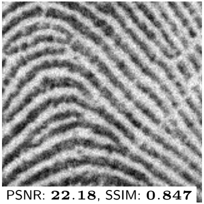
Non-Local Means

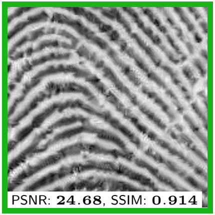


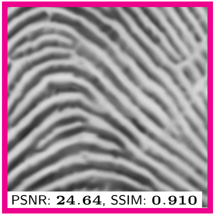
CLEAR

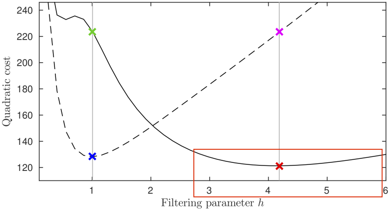
Bias-variance trade-off: Non-Local Means

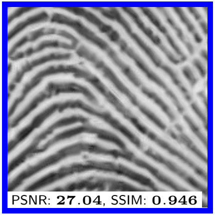




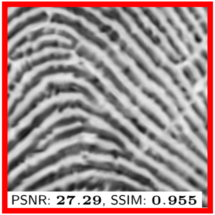








Non-Local Means



CLEAR

2 steps Implementation for Black Box algorithm

- Denoising algorithm: $y \mapsto \hat{x}(y)$
- Re-fitting with finite difference:

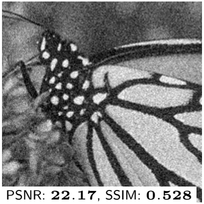
  1: $\delta = y - \hat{x}(y)$

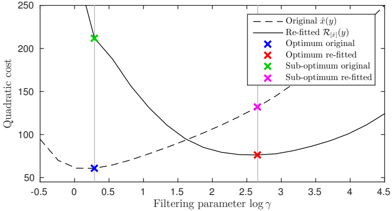
  2: $J\delta = \frac{\hat{x}(y+\varepsilon\delta) - \hat{x}(y))}{\varepsilon}$

  3: $\rho = \frac{\langle J\delta, \delta \rangle}{||J\delta||_2^2}$

  4: $\mathcal{R}_{\hat{x}}^{\mathrm{cov}}(y) = \hat{x}(y) + \rho J\delta$

BM3D [Dabov et al. 2007, Lebrun 2012]

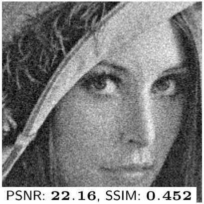


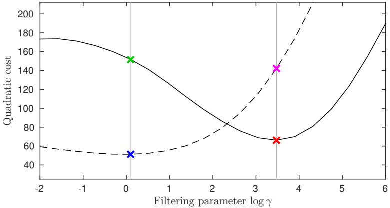


BM3D

CLEAR

DDID [Knaus & Zwicker 2013]





DDID

CLEAR

DnCNN [Zhang et al., 2017]

Residual Network learning noise to remove

Noise level 25 DnCNN

DnCNN [Zhang et al., 2017]

Residual Network learning noise to remove

Noise level 25 CLEAR

DnCNN [Zhang et al., 2017]

Residual Network learning noise to remove

Noise level 50 DnCNN

DnCNN [Zhang et al., 2017]

Residual Network learning noise to remove

Noise level 50 CLEAR

DnCNN [Zhang et al., 2017]

Residual Network learning noise to remove

Noise level 150 DnCNN

DnCNN [Zhang et al., 2017]

Residual Network learning noise to remove

Noise level 150 CLEAR

DnCNN [Zhang et al., 2017]

Residual Network learning noise to remove

Noise level 150 CLEAR

- No interesting structural information to recover from noise model

FFDNet [Zhang et al., 2018]

Network learning denoised image for Gaussian noise of variance $[0;75]$

Noise level 25 FFDNet

FFDNet [Zhang et al., 2018]

Network learning denoised image for Gaussian noise of variance $[0; 75]$

Noise level 25 CLEAR

FFDNet [Zhang et al., 2018]

Network learning denoised image for Gaussian noise of variance $[0; 75]$

Noise level 50 FFDNet

FFDNet [Zhang et al., 2018]

Network learning denoised image for Gaussian noise of variance $[0; 75]$

Noise level 50 CLEAR

FFDNet [Zhang et al., 2018]

Network learning denoised image for Gaussian noise of variance $[0; 75]$

Noise level 150 FFDNet

FFDNet [Zhang et al., 2018]

Network learning denoised image for Gaussian noise of variance $[0; 75]$

Noise level 150 CLEAR

Conclusions

Covariant LEAst-square Re-fitting

- Correct part of the bias of restoration models
- No aditionnal parameter
- Stability for a larger range of parameters
- Double the computational cost

Conclusions

Covariant LEAst-square Re-fitting

- Correct part of the bias of restoration models
- No aditionnal parameter
- Stability for a larger range of parameters
- Double the computational cost

When using re-fitting?

- Differentiable estimators: no algorithm with quantization
- Regularization prior adapted to data
- Respect data range: oceanography, radiotherapy...

Main related references

[Brinkmann, Burger, Rasch and Sutour] Bias-reduction in variational regularization.*JMIV*, 2017.

[Deledalle, P. and Salmon] On debiasing restoration algorithms: applications to total-variation and nonlocal-means. *SSVM*, 2015.

[Deledalle, P., Salmon and Vaiter] CLEAR: Covariant LEAst-square Re-fitting. *SIAM SIIMS*, 2017.

[Deledalle, P., Salmon and Vaiter] Refitting solutions with block penalties, *SSVM*, 2019.

[Osher, Burger, Goldfarb, Xu, and Yin] An iterative regularization method for total variation-based image restoration. *SIAM MMS*, 2005

[Romano and Elad] Boosting of image denoising algorithms. *SIAM SIIMS*, 2015.

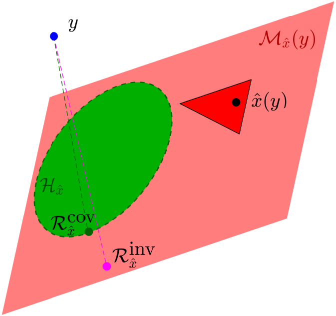
[Talebi, Zhu and Milanfar] How to SAIF-ly boost denoising performance. *IEEE TIP*, 2013.

[Vaiter, Deledalle, Peyré, Fadili and Dossal] The degrees of freedom of partly smooth regularizers. *Annals of the Institute of Statistical Mathematics*, 2016.

Sign in TV models [Brinkmann et al. 2016]

- Differentiable estimator w.r.t y:

$$\hat{x}(y) = \underset{x}{\operatorname{argmin}} \|x - y\|^2 + \lambda \|\nabla x\|_{1,2}$$



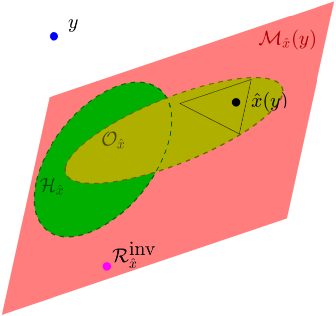
Sign in TV models [Brinkmann et al. 2016]

- Differentiable estimator w.r.t y:

$$\hat{x}(y) = \underset{x}{\operatorname{argmin}} \|x - y\|^2 + \lambda \|\nabla x\|_{1,2}$$

- Orientation preservation:

$$\mathcal{O}_{\hat{x}(y)} = \{x | \, (\nabla x)_i = \alpha_i (\nabla \hat{x})_i, \, \forall i \in \mathcal{I}\}$$



Sign in TV models [Brinkmann et al. 2016]

- Differentiable estimator w.r.t y:

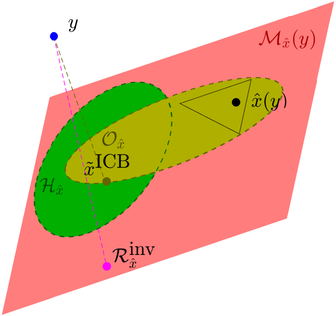
$$\hat{x}(y) = \underset{x}{\operatorname{argmin}} \|x - y\|^2 + \lambda \|\nabla x\|_{1,2}$$

- Orientation preservation:

$$\mathcal{O}_{\hat{x}(y)} = \{x | \, (\nabla x)_i = \alpha_i (\nabla \hat{x})_i, \, \forall i \in \mathcal{I}\}$$

- Infimal Convolution of Bregman divergences

$$\tilde{x}^{\mathrm{ICB}} = \underset{x \in \mathcal{O}_{\hat{x}(y)}}{\operatorname{argmin}} \|x - y\|^2$$



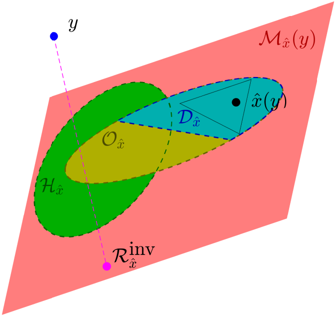
Sign in TV models [Brinkmann et al. 2016]

- Differentiable estimator w.r.t y:

$$\hat{x}(y) = \underset{x}{\operatorname{argmin}} \|x - y\|^2 + \lambda \|\nabla x\|_{1,2}$$

- Direction preservation:

$$\mathcal{D}_{\hat{x}(y)} = \{x | \, (\nabla x)_i = \alpha_i (\nabla \hat{x})_i, \, \alpha_i \geqslant 0, \, \forall i \in \mathcal{I}\}$$



Sign in TV models [Brinkmann et al. 2016]

- Differentiable estimator w.r.t y:

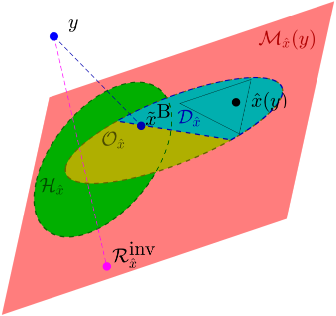
$$\hat{x}(y) = \underset{x}{\operatorname{argmin}} \|x - y\|^2 + \lambda \|\nabla x\|_{1,2}$$

- Direction preservation:

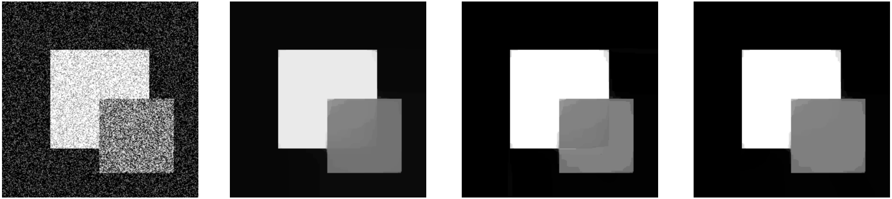
$$\mathcal{D}_{\hat{x}(y)} = \{x | \, (\nabla x)_i = \alpha_i (\nabla \hat{x})_i, \, \alpha_i \geqslant 0, \, \forall i \in \mathcal{I}\}$$

- Bregman divergence

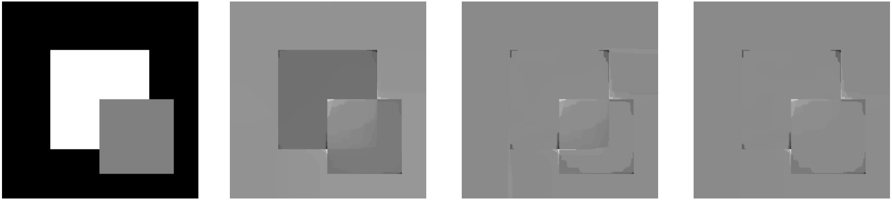
$$\tilde{x}^{\mathrm{B}} = \underset{x \in \mathcal{D}_{\hat{x}(y)}}{\operatorname{argmin}} \|x - y\|^2$$



Sign influence in Re-fitting



y | Anisotropic TV $\hat{x}(y)$ | $\mathcal{R}^{\mathrm{inv}}_{\hat{x}} = \mathcal{R}^{\mathrm{cov}}_{\hat{x}} = \tilde{x}^{\mathrm{ICB}}$ Orientation [Brinkmann et al.] | $\tilde{x}^{\mathrm{B}}$ Direction [Brinkmann et al.]

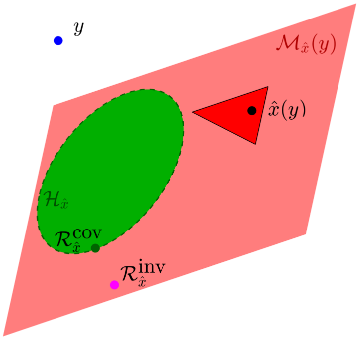


x_0 | $\hat{x}(y) - x_0$ | $\tilde{x}^{\mathrm{ICB}} - x_0$ | $\tilde{x}^{\mathrm{B}} - x_0$

New Re-fitting models

- Covariant Re-fitting:

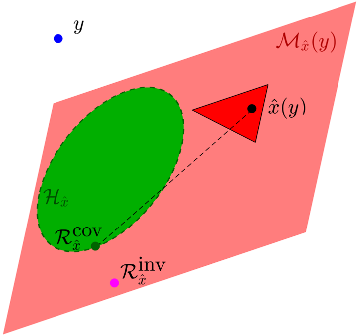
$$\mathcal{R}_{\hat{x}}^{\text{cov}}(y) = \underset{x \in \mathcal{H}_{\hat{x}}}{\text{argmin}} \ \frac{1}{2}||\Phi x - y||^2$$



New Re-fitting models

- Covariant Re-fitting:

$$\mathcal{R}^{\text{cov}}_{\hat{x}}(y) = \operatorname*{argmin}_{x \in \mathcal{H}_{\hat{x}}} \frac{1}{2}||\Phi x - y||^2 = \hat{x}(y) + \rho J(y - \Phi \hat{x}(y))$$



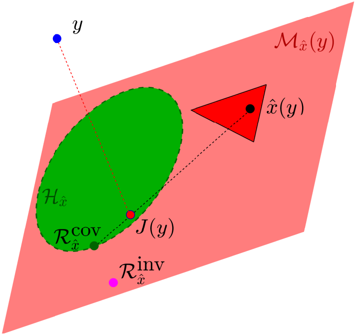
New Re-fitting models

- Covariant Re-fitting:

$$\mathcal{R}^{\text{cov}}_{\hat{x}}(y) = \underset{x \in \mathcal{H}_{\hat{x}}}{\operatorname{argmin}} \ \frac{1}{2}||\Phi x - y||^2 = \hat{x}(y) + \rho J(y - \Phi \hat{x}(y))$$

- Apply linear Jacobian: orientation penalization

$$J(y) = \underset{x \in \mathcal{M}_{\hat{x}}(y)}{\operatorname{argmin}} \ \frac{1}{2}||\Phi x - y||^2$$
$$+ \underbrace{\left(\frac{\lambda}{2||\nabla \hat{x}||} \left|\left| \nabla x - \langle \nabla \hat{x}, \nabla x \rangle \frac{\nabla \hat{x}}{||\nabla \hat{x}||^2} \right|\right|^2 \right)_{\mathcal{I}}}_{=0,\ \forall x \in \mathcal{O}_{\hat{x}}(y)}$$



New Re-fitting models

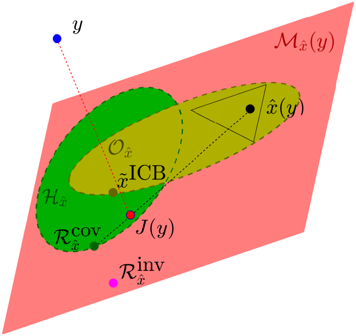
- Covariant Re-fitting:

$$\mathcal{R}^{\text{cov}}_{\hat{x}}(y) = \underset{x \in \mathcal{H}_{\hat{x}}}{\operatorname{argmin}} \frac{1}{2}||\Phi x - y||^2 = \hat{x}(y) + \rho J(y - \Phi \hat{x}(y))$$

- Apply linear Jacobian: orientation penalization

$$J(y) = \underset{x \in \mathcal{M}_{\hat{x}}(y)}{\operatorname{argmin}} \frac{1}{2}||\Phi x - y||^2 + \underbrace{\left(\frac{\lambda}{2||\nabla \hat{x}||}\left|\left|\nabla x - \langle \nabla \hat{x}, \nabla x\rangle \frac{\nabla \hat{x}}{||\nabla \hat{x}||^2}\right|\right|^2\right)_{\mathcal{I}}}_{=0,\, \forall x \in \mathcal{O}_{\hat{x}}(y)}$$

- $J(y)$: Compromise between $\tilde{x}^{\text{ICB}}$ and $\mathcal{R}^{\text{inv}}_{\hat{x}}$



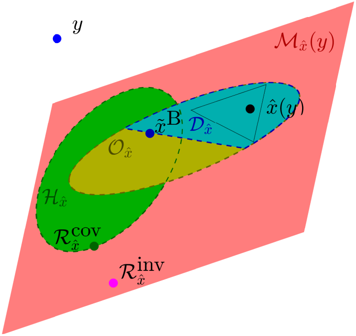
New Re-fitting models

- Covariant Re-fitting:

$$\mathcal{R}^{\text{cov}}_{\hat{x}}(y) = \underset{x\in\mathcal{H}_{\hat{x}}}{\operatorname{argmin}} \frac{1}{2}||\Phi x - y||^2 = \hat{x}(y) + \rho J(y - \Phi\hat{x}(y))$$

- Apply linear Jacobian: orientation penalization

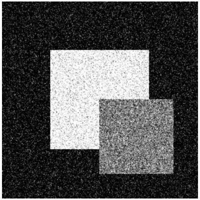
$$J(y) = \underset{x\in\mathcal{M}_{\hat{x}}(y)}{\operatorname{argmin}} \frac{1}{2}||\Phi x - y||^2 + \underbrace{\left(\frac{\lambda}{2||\nabla\hat{x}||}\left|\left|\nabla x - \langle\nabla\hat{x}, \nabla x\rangle\frac{\nabla\hat{x}}{||\nabla\hat{x}||^2}\right|\right|^2\right)_{\mathcal{I}}}_{=0,\ \forall x\in\mathcal{O}_{\hat{x}}(y)}$$



- $J(y)$: Compromise between $\tilde{x}^{\text{ICB}}$ and $\mathcal{R}^{\text{inv}}_{\hat{x}}$
- New Re-fitting penalizing direction changes

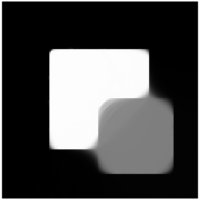
$$\mathcal{R}_{\hat{x}}(y) = \underset{x\in\mathcal{M}_{\hat{x}}(y)}{\operatorname{argmin}} \frac{1}{2}||\Phi x - y||^2 + \underbrace{F(\nabla x, \nabla\hat{x}(y))_{\mathcal{I}}}_{=0,\ \forall x\in\mathcal{D}_{\hat{x}}(y)}$$

Comparison of Re-fitting approaches

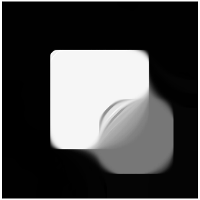


y

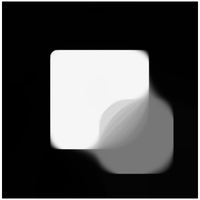
Tv iso



Bregman iteration
[Osher et al.]

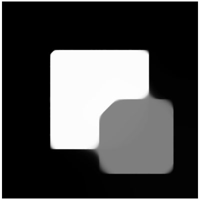


Orientation
[Brinkmann et al.]



Direction
[Brinkmann et al.]

Covariant



New model

Comparison of Re-fitting approaches

y

Tv iso

Bregman iteration
[Osher et al.]

Orientation
[Brinkmann et al.]

Direction
[Brinkmann et al.]

Covariant

New model