CLEAR: Covariant Least-Square Refitting

Charles DELEDALLE, Nicolas PAPADAKIS, Joseph SALMON &
Samuel VAITER

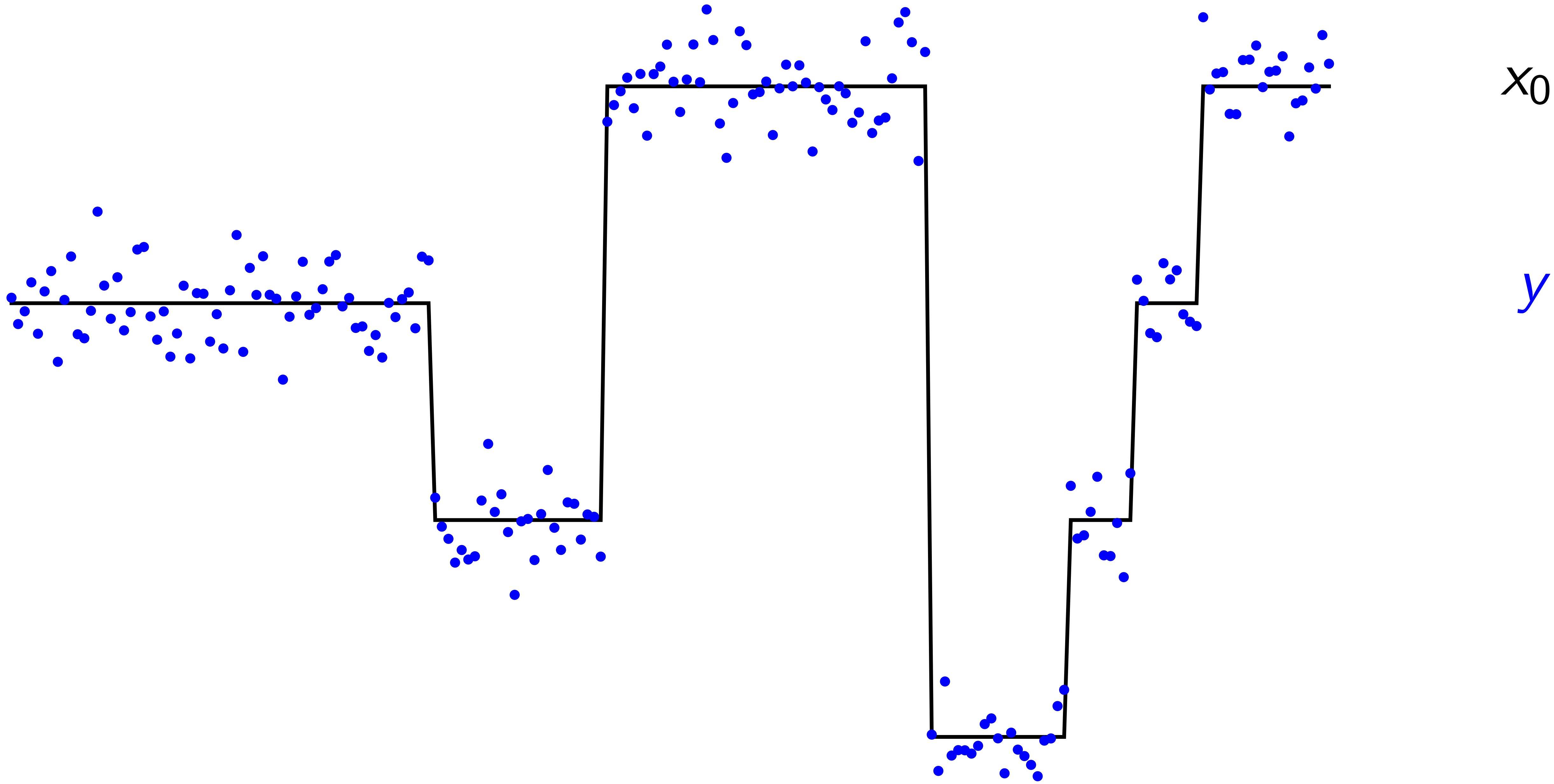
CNRS & IMB, Dijon, France

16/07/2019
ICIAM'19

A Starting Point

Noisy observations

$$y = x_0 + w$$



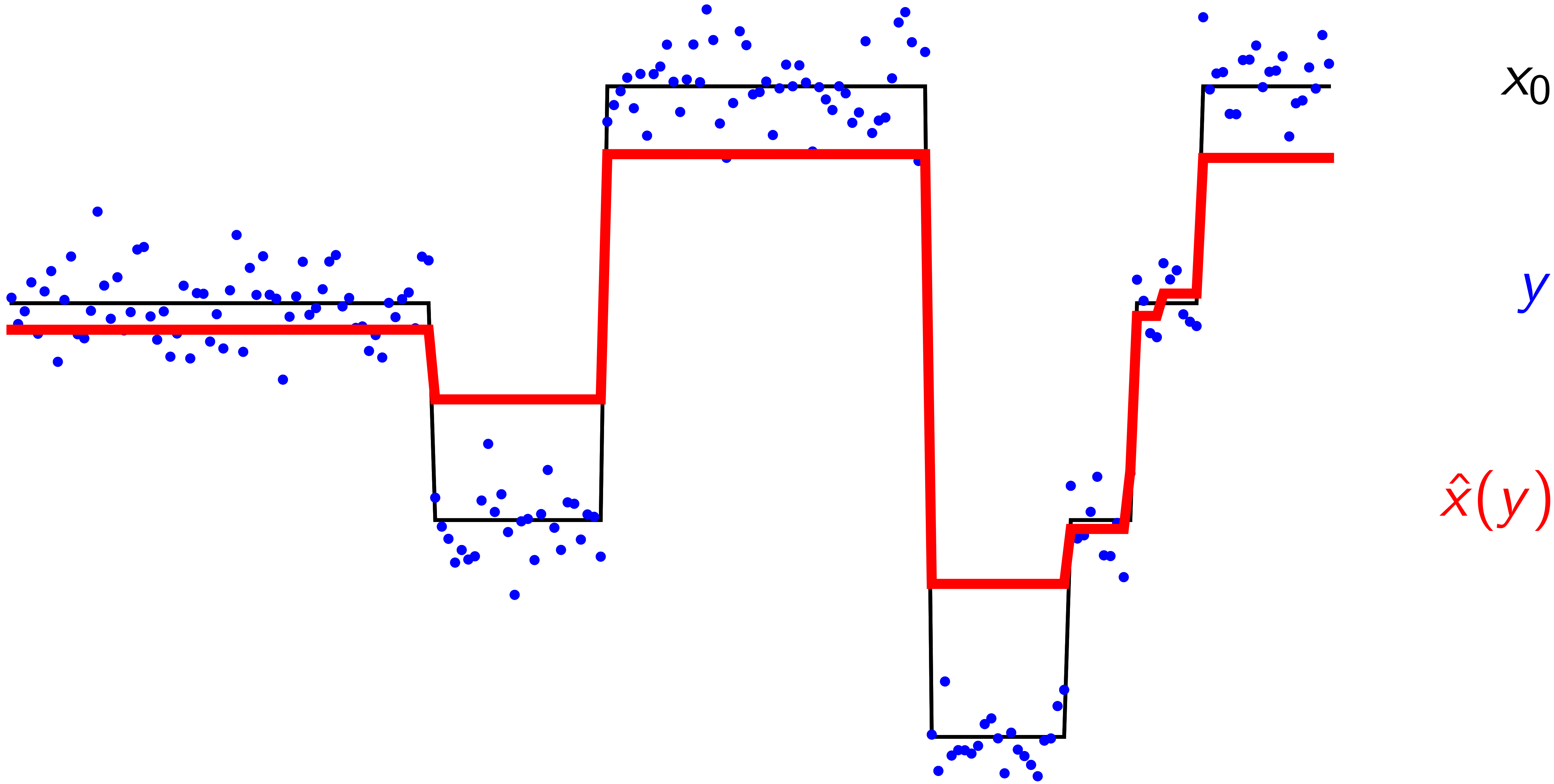
A Starting Point

Noisy observations

$$y = x_0 + w$$

Total Variation regularization [Rudin et al. 92]

$$\hat{x}(y) = \underset{x \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{2}\|x - y\|_2^2 + \lambda \|\nabla x\|_1$$



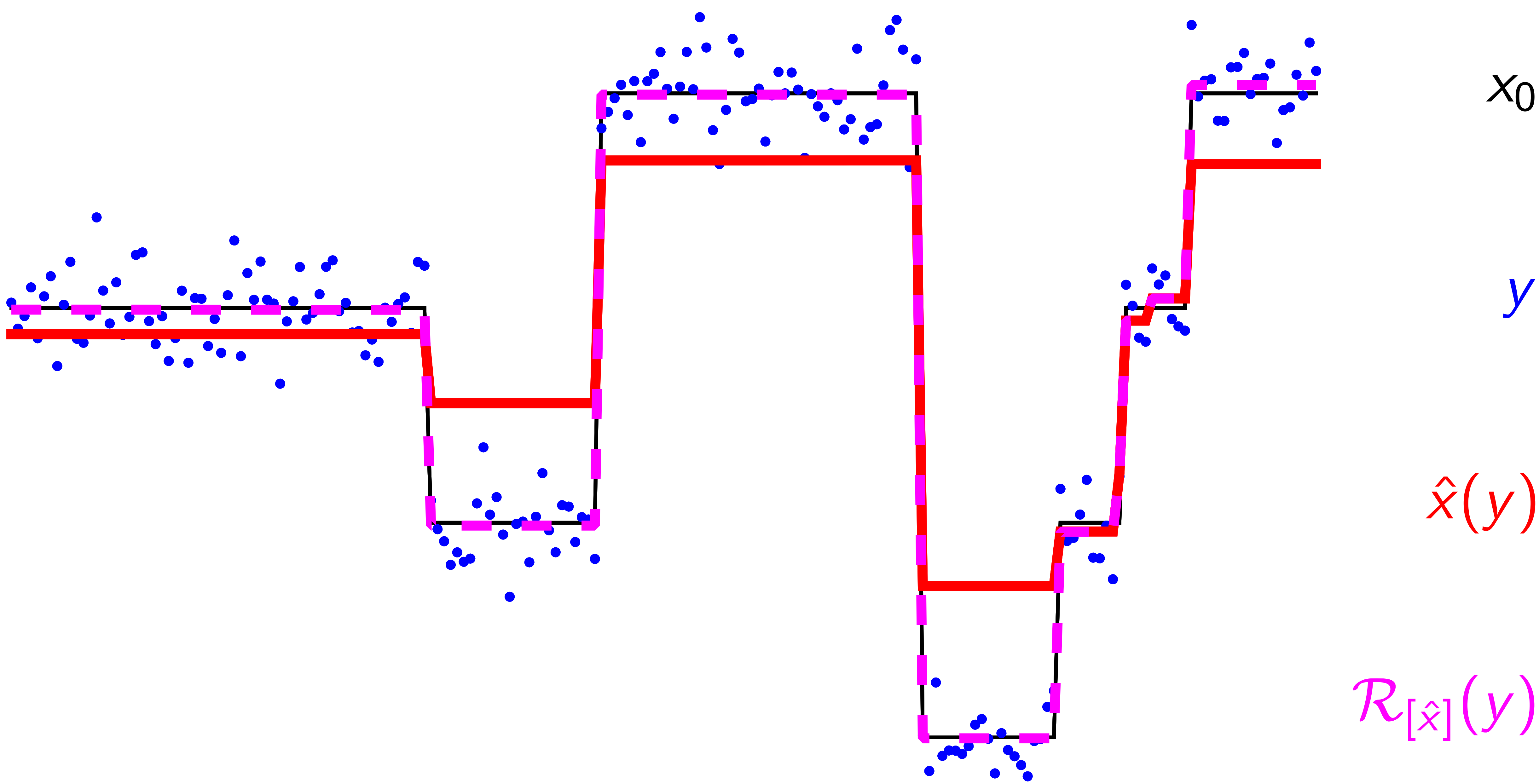
A Starting Point

Noisy observations

$$y = x_0 + w$$

Total Variation regularization [Rudin et al. 92]

$$\hat{x}(y) = \underset{x \in \mathbb{R}^p}{\operatorname{argmin}} \ \frac{1}{2}\|x - y\|_2^2 + \lambda \|\nabla x\|_1$$



Generalization of the Mean-Jump

Preserving the subspace $\{x \ : \ \mathrm{supp}(\nabla x) = \mathrm{supp}(\nabla \hat{x}(y))\}$

$\longrightarrow$ refitting [Efron et al. 2004], [Lederer 2013]

Generalization of the Mean-Jump

Preserving the subspace $\{x \ : \ \mathrm{supp}(\nabla x) = \mathrm{supp}(\nabla \hat{x}(y))\}$

$\rightarrow$ refitting [Efron et al. 2004], [Lederer 2013]

General point of view ?

Generalization of the Mean-Jump

Preserving the subspace $\{x \;:\; \text{supp}(\nabla x) = \text{supp}(\nabla \hat{x}(y))\}$

$\longrightarrow$ refitting [Efron et al. 2004], [Lederer 2013]

General point of view ?

TV

Model space (for a weakly differentiable estimator)

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

Jacobian of $\hat{x}$ at y

Generalization of the Mean-Jump

Preserving the subspace $\{x \;:\; \mathrm{supp}(\nabla x) = \mathrm{supp}(\nabla \hat{x}(y))\}$

$\to$ refitting [Efron et al. 2004], [Lederer 2013]

General point of view ?

TV

Model space (for a weakly differentiable estimator)

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

Jacobian of $\hat{x}$ at y

Invariant Re-enhancement [Deledalle, Papadakis & Salmon 2015]

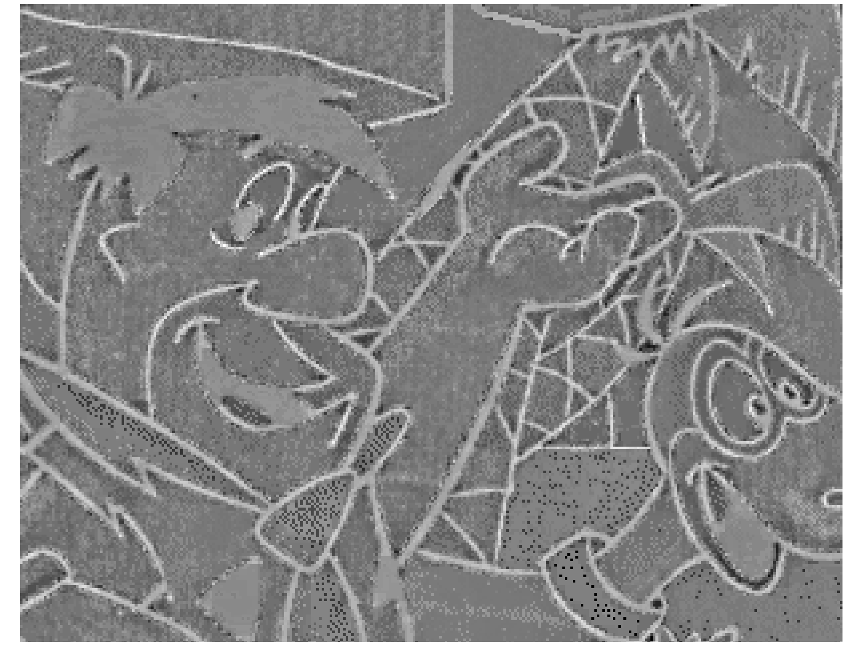
$$\mathcal{R}^{\mathrm{inv}}_{[\hat{x}]}(y) = \underset{x \in \mathcal{M}_{[\hat{x}]}(y)}{\mathrm{argmin}} \; \|y - x\|_2^2$$

$$\mathcal{R}^{\mathrm{inv}}_{[\hat{x}]}(y) = \hat{x}(y) + JJ^{+}(y - \hat{x}(y))$$

Performance on Anisotropic TV

y

$\hat{x}(y)$

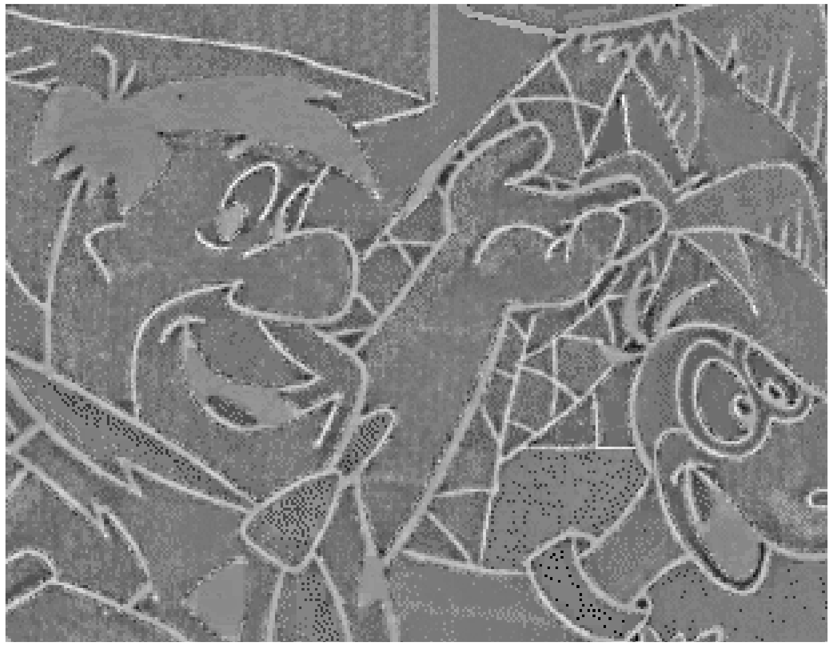


$\hat{x}(y) - x_0$

Performance on Anisotropic TV

y

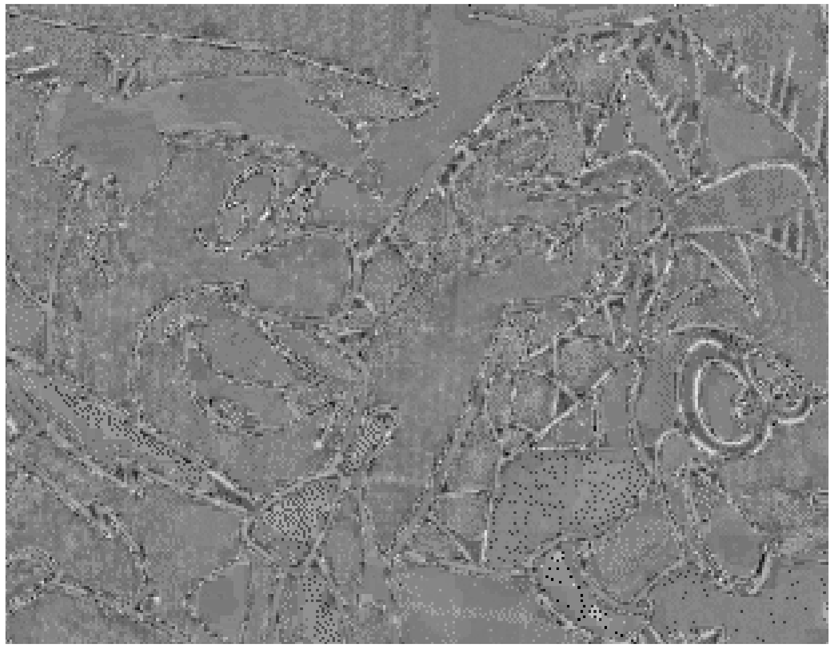
$\hat{x}(y)$



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

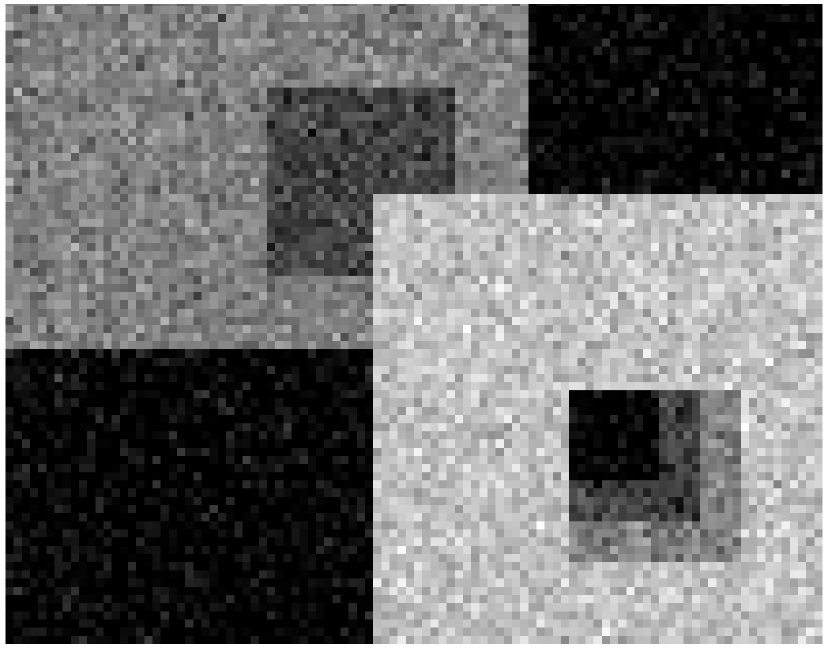
$$\mathcal{R}^{\text{inv}}_{[\hat{x}]}(y) = J_{\hat{x}}(y)$$

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



$\mathcal{R}^{\text{inv}}_{[\hat{x}]}(y) - x_0$

Performance on Isotropic TV

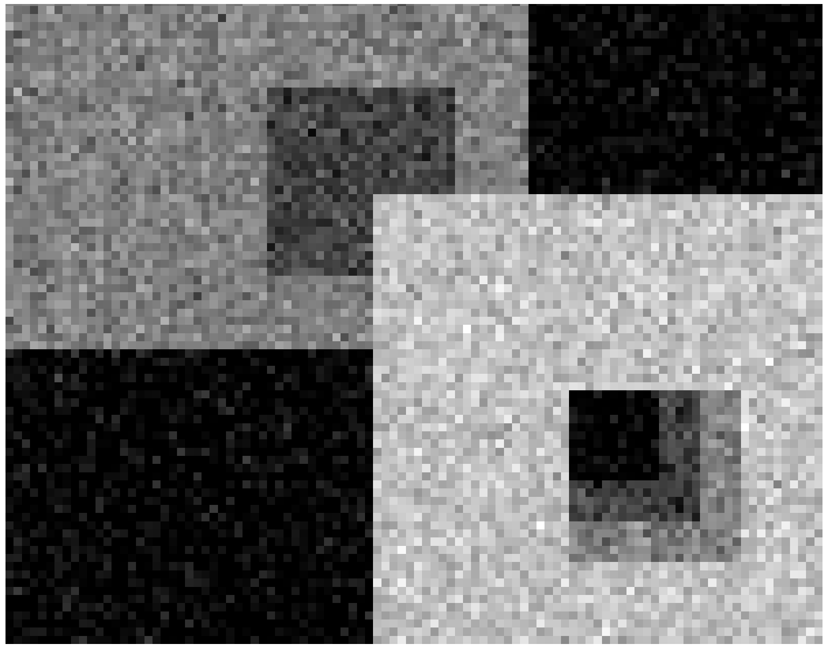


y

$\hat{x}(y)$

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

Performance on Isotropic TV



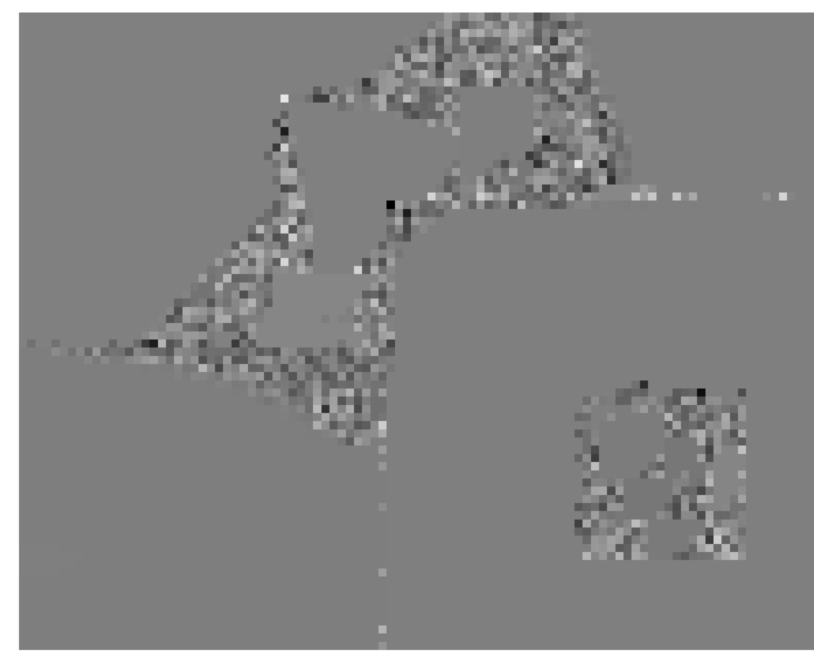
y

$\hat{x}(y)$

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

$\mathcal{R}^{\mathrm{inv}}_{[\hat{x}]}(y) \neq J_{\hat{x}}(y)$

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



$\mathcal{R}^{\mathrm{inv}}_{[\hat{x}]}(y) - x_0$

From Invariant to Covariant Renhancement

the model space captures only linear invariance

$$\mathcal{R}^{\text{inv}}_{[\hat{x}]}(y) = \underset{x \in \mathcal{M}_{[\hat{x}]}(y)}{\text{argmin}} \; \|y - x\|_2^2$$

↓

need more constraints

From Invariant to Covariant Renhancement

the model space captures only linear invariance

$$\mathcal{R}^{\text{inv}}_{[\hat{x}]}(y) = \underset{x \in \mathcal{M}_{[\hat{x}]}(y)}{\text{argmin}} \; \|y - x\|_2^2$$

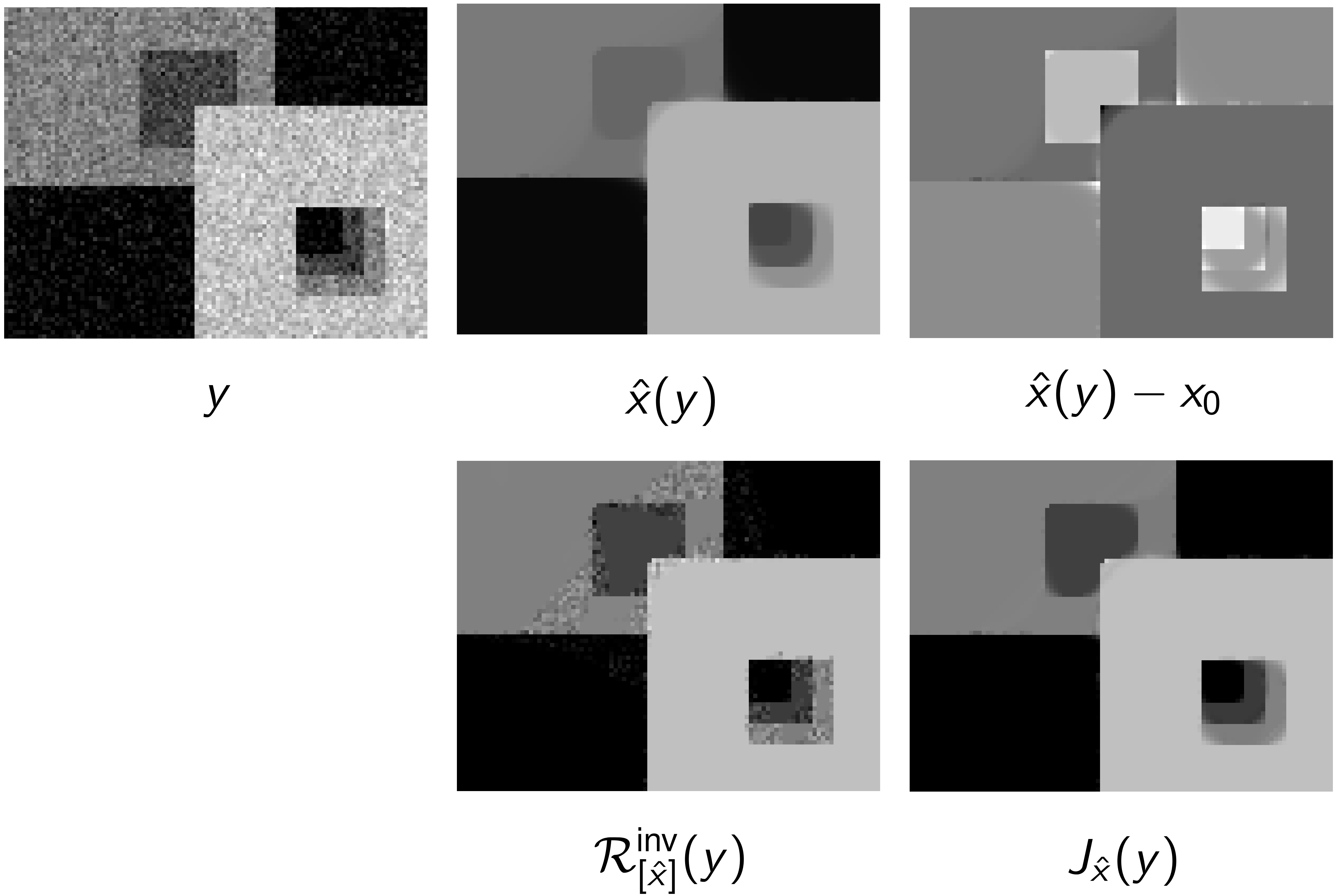
↓

need more constraints

Empirical observation

The Jacobian captures more invariances

From Invariant to Covariant Renhancement



Local Approach

$$\hat{x} : \mathbb{R}^p \to \mathbb{R}^p$$

$$y \mapsto \hat{x}(y)$$

Local Approach

$$\hat{x} : \mathbb{R}^p \to \mathbb{R}^p$$
$$y \mapsto \hat{x}(y)$$

enhance locally

$$\mathcal{D} : ((\mathbb{R}^p \to \mathbb{R}^p) \times \mathbb{R}^p) \to (\mathbb{R}^p \to \mathbb{R}^p)$$
$$(\hat{x}, y) \mapsto \mathcal{D}_{\hat{x}(y)}$$

Local Approach

$$\hat{x} : \mathbb{R}^p \to \mathbb{R}^p$$
$$y \mapsto \hat{x}(y)$$

enhance locally

$$\mathcal{D} : ((\mathbb{R}^p \to \mathbb{R}^p) \times \mathbb{R}^p) \to (\mathbb{R}^p \to \mathbb{R}^p)$$
$$(\hat{x}, y) \mapsto \mathcal{D}_{\hat{x}(y)}$$

apply point-wise

$$\mathcal{R}_{[\hat{x}]} : \mathbb{R}^p \to \mathbb{R}^p$$
$$y \mapsto \mathcal{R}_{[\hat{x}]}(y) = \mathcal{D}_{\hat{x}(y)}(y)$$

Covariant Re-enhancement

Local constraints

Affine map	$\mathcal{D}_{\hat{x}(y)}(z) = Az + b$
Covariant preserving	$J_{\mathcal{D}_{\hat{x}(y)}}(y) = J_{\hat{x}}(y)$
Coherent map	$\mathcal{D}_{\hat{x}(y)}(\hat{x}(y)) = \hat{x}(y)$

Covariant Re-enhancement

Local constraints

Affine map	$\mathcal{D}_{\hat{x}(y)}(z) = Az + b$
Covariant preserving	$J_{\mathcal{D}_{\hat{x}(y)}}(y) = J_{\hat{x}}(y)$
Coherent map	$\mathcal{D}_{\hat{x}(y)}(\hat{x}(y)) = \hat{x}(y)$

Characterization

$$\mathcal{D}_{\hat{x}(y)}(z) = \hat{x}(y) + J(z - \hat{x}(y))$$

Covariant Re-enhancement

Local constraints

Affine map	$\mathcal{D}_{\hat{x}(y)}(z) = Az + b$
Covariant preserving	$J_{\mathcal{D}_{\hat{x}(y)}}(y) = J_{\hat{x}}(y)$
Coherent map	$\mathcal{D}_{\hat{x}(y)}(\hat{x}(y)) = \hat{x}(y)$

Characterization

$$\mathcal{D}_{\hat{x}(y)}(z) = \hat{x}(y) + J(z - \hat{x}(y))$$

Definition

$$\mathcal{R}_{[\hat{x}]}(y) = \mathcal{D}_{\hat{x}(y)}(y)$$

$$\mathcal{R}_{[\hat{x}]}(y) = \hat{x}(y) + J(y - \hat{x}(y))$$

Covariant Re-enhancement

Local constraints

Affine map $\mathcal{D}_{\hat{x}(y)}(z) = Az + b$

Covariant preserving $J_{\mathcal{D}_{\hat{x}(y)}}(y) = J_{\hat{x}}(y)$

Coherent map $\mathcal{D}_{\hat{x}(y)}(\hat{x}(y)) = \hat{x}(y)$

Characterization

$$\mathcal{D}_{\hat{x}(y)}(z) = \hat{x}(y) + J(z - \hat{x}(y))$$

Definition

$$\mathcal{R}_{[\hat{x}]}(y) = \mathcal{D}_{\hat{x}(y)}(y)$$

$$\mathcal{R}_{[\hat{x}]}(y) = \hat{x}(y) + J(y - \hat{x}(y))$$

Invariant vs Covariant

$$\mathcal{R}^{\text{inv}}_{[\hat{x}]}(y) = \hat{x}(y) + JJ^{+}(y - \hat{x}(y)) \qquad \mathcal{R}_{[\hat{x}]}(y) = \hat{x}(y) + J(y - \hat{x}(y))$$

Covariant Re-enhancement

Local constraints

Affine map $\mathcal{D}_{\hat{x}(y)}(z) = Az + b$

Covariant preserving $J_{\mathcal{D}_{\hat{x}(y)}}(y) = J_{\hat{x}}(y)$

Coherent map $\mathcal{D}_{\hat{x}(y)}(\hat{x}(y)) = \hat{x}(y)$

Characterization

$$\mathcal{D}_{\hat{x}(y)}(z) = \hat{x}(y) + J(z - \hat{x}(y))$$

Definition

$$\mathcal{R}_{[\hat{x}]}(y) = \mathcal{D}_{\hat{x}(y)}(y)$$

$$\mathcal{R}_{[\hat{x}]}(y) = \hat{x}(y) + J(y - \hat{x}(y))$$

Invariant vs Covariant

$$\mathcal{R}^{\text{inv}}_{[\hat{x}]}(y) = \hat{x}(y) + JJ^{+}(y - \hat{x}(y)) \qquad \mathcal{R}_{[\hat{x}]}(y) = \hat{x}(y) + J(y - \hat{x}(y))$$

Naive point of view: just remove the pseudo-inverse!

Covariant Re-enhancement

Local constraints

Affine map	$\mathcal{D}_{\hat{x}(y)}(z) = Az + b$
Covariant preserving	$J_{\mathcal{D}_{\hat{x}(y)}}(y) = \rho J_{\hat{x}}(y)$
Coherent map	$\mathcal{D}_{\hat{x}(y)}(\hat{x}(y)) = \hat{x}(y)$

Characterization

$$\mathcal{D}_{\hat{x}(y)}(z) = \hat{x}(y) + \rho J(z - \hat{x}(y))$$

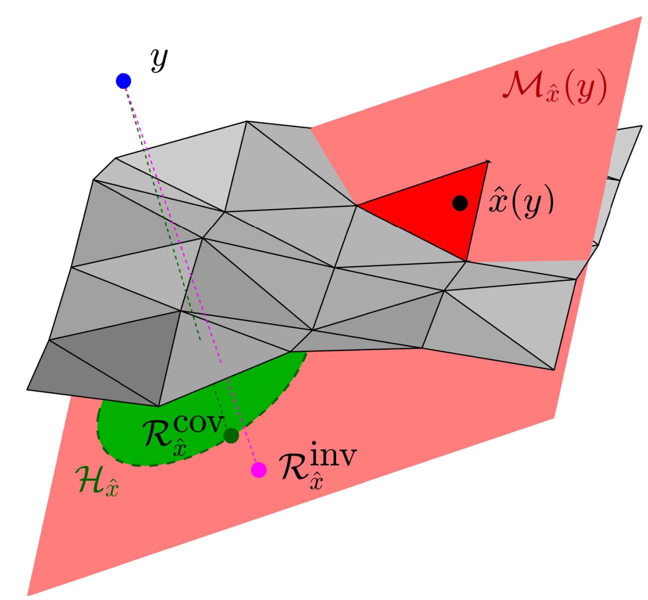
Definition

$$\mathcal{R}_{[\hat{x}]}(y) = \mathcal{D}_{\hat{x}(y)}(y)$$

$$\mathcal{R}_{[\hat{x}]}(y) = \hat{x}(y) + \rho J(y - \hat{x}(y))$$

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

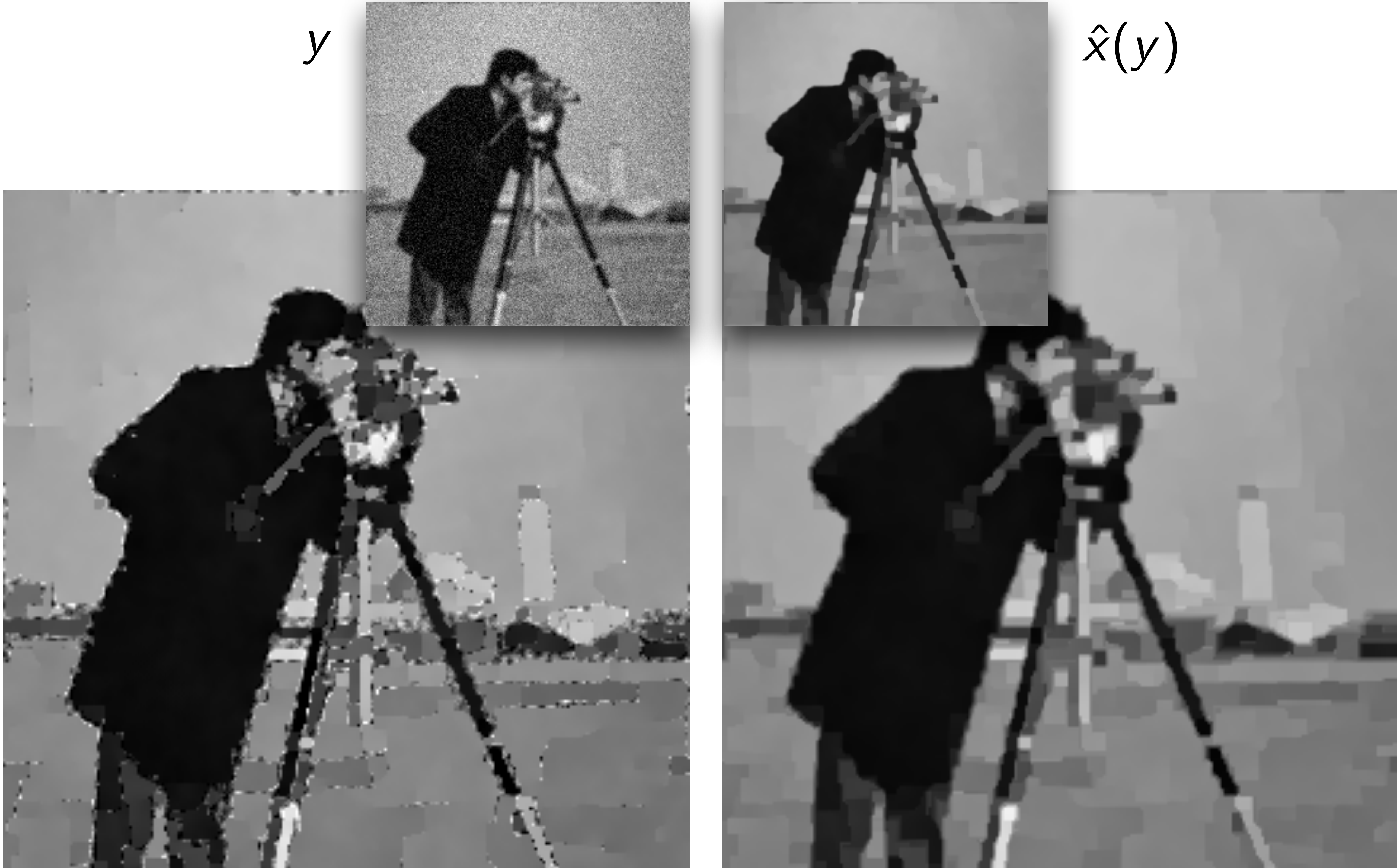
$$\rho = \begin{cases} \frac{\langle J\delta, \delta \rangle}{\|J\delta\|^2} & \text{if } J\delta \neq 0 \\ 1 & \text{otherwise} \end{cases}$$



Numerical Stability

y

$\hat{x}(y)$



$\mathcal{R}_{[\hat{x}^k]}(y)$

$\mathcal{R}_{[\hat{x}]}(y)$

Numerical Stability

$$\hat{x}(y) = \underset{x \in \mathbb{R}^p}{\text{argmin}} \ \frac{1}{2}\|x - y\|_2^2 + \lambda \|\nabla x\|_1$$

Computing $\mathcal{R}_{[\hat{x}]}(y)$ requires the knowledge of $\text{supp}(\nabla \hat{x}(y))$

But in practice, $\hat{x}(y)$ is approximated through a sequence $\hat{x}^k(y)$

Unfortunately, $\hat{x}^k(y) \approx \hat{x}(y) \not\Rightarrow \text{supp}(\hat{x}^k(y)) \approx \text{supp}(\hat{x}(y))$

y $\quad$ $\hat{x}(y)$ $\quad$ $\mathcal{R}_{[\hat{x}^k]}(y)$ $\quad$ $\mathcal{R}_{[\hat{x}]}(y)$

Joint Estimation—Re-enhancement

$$\mathcal{R}_{[\hat{x}]}(y) = \hat{x}(y) + J(y - \hat{x}(y))$$

Iterative algorithm

$$\hat{x}^{k+1}(y) = \mathcal{A}_k(\hat{x}^k(y), y)$$

Joint Estimation — Re-enhancement

$$\mathcal{R}_{[\hat{x}]}(y) = \hat{x}(y) + J(y - \hat{x}(y))$$

Iterative algorithm

$$\hat{x}^{k+1}(y) = \mathcal{A}_k(\hat{x}^k(y), y)$$

Chained expression

$$\hat{x}^k(y) = \mathcal{A}_k \circ \mathcal{A}_{k-1} \circ \cdots \circ \mathcal{A}_0(\hat{x}^0(y), y)$$

Joint Estimation — Re-enhancement

$$\mathcal{R}_{[\hat{x}]}(y) = \hat{x}(y) + J(y - \hat{x}(y))$$

Iterative algorithm

$$\hat{x}^{k+1}(y) = \mathcal{A}_k(\hat{x}^k(y), y)$$

Chained expression

$$\hat{x}^k(y) = \mathcal{A}_k \circ \mathcal{A}_{k-1} \circ \cdots \circ \mathcal{A}_0(\hat{x}^0(y), y)$$

Chain rule

$$\frac{\partial}{\partial y}\mathcal{A}_k \circ \frac{\partial}{\partial y}\mathcal{A}_{k-1} \circ \cdots \circ \frac{\partial}{\partial y}\mathcal{A}_0(y) \overset{*}{\to} Jy$$

*Unfortunately, to prove this convergence, further assumptions have to be made (convergence of functions does not impl convergence of derivatives…). It works for instance for anisotropic TV.

One-step vs Two-step Evaluation

Two step

$$\frac{\partial}{\partial y}\mathcal{A}_k \circ \frac{\partial}{\partial y}\mathcal{A}_{k-1} \circ \cdots \circ \frac{\partial}{\partial y}\mathcal{A}_0(y) \to Jy$$

$$\frac{\partial}{\partial y}\mathcal{A}_k \circ \frac{\partial}{\partial y}\mathcal{A}_{k-1} \circ \cdots \circ \frac{\partial}{\partial y}\mathcal{A}_0(\hat{x}(y)) \to J\hat{x}(y)$$

$$\mathcal{R}_{[\hat{x}]}(y) = \hat{x}(y) + J(y - \hat{x}(y))$$

One-step vs Two-step Evaluation

Two step

$$\frac{\partial}{\partial y}\mathcal{A}_k \circ \frac{\partial}{\partial y}\mathcal{A}_{k-1} \circ \cdots \circ \frac{\partial}{\partial y}\mathcal{A}_0(y) \to Jy$$

$$\frac{\partial}{\partial y}\mathcal{A}_k \circ \frac{\partial}{\partial y}\mathcal{A}_{k-1} \circ \cdots \circ \frac{\partial}{\partial y}\mathcal{A}_0(\hat{x}(y)) \to J\hat{x}(y)$$

$$\mathcal{R}_{[\hat{x}]}(y) = \hat{x}(y) + J(y - \hat{x}(y))$$

One step

$$\text{If } J\hat{x}(y) = \hat{x}(y) \Longrightarrow \mathcal{R}_{[\hat{x}]}(y) = Jy$$

$\to$ theoretical general results

$\to$ true for aniso-TV, iso-TV, Lasso, ...

Example for Anisotropic TV

$$\hat{x}(y) = \underset{x \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{2}\|x - y\|_2^2 + \lambda\|\nabla x\|_1$$

$$\begin{aligned} z^{k+1} &= \Pi_{B_\lambda}(z^k + \sigma \nabla v^k) \\ x^{k+1} &= (1+\tau)^{-1}\left(x^k + \tau(y + \operatorname{div} z^{k+1})\right) \\ v^{k+1} &= x^{k+1} + \theta(x^{k+1} - x^k) \end{aligned}$$

Chambolle-Pock

Π_{B_λ} projection on the λ-ball

Example for Anisotropic TV

$$\hat{x}(y) = \underset{x \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{2}\|x - y\|_2^2 + \lambda \|\nabla x\|_1$$

$$z^{k+1} = \Pi_{B_\lambda}(z^k + \sigma \nabla v^k)$$

$$x^{k+1} = (1+\tau)^{-1}\left(x^k + \tau(y + \operatorname{div} z^{k+1})\right)$$

$$v^{k+1} = x^{k+1} + \theta(x^{k+1} - x^k)$$

Chambolle-Pock

$$\tilde{z}^{k+1} = \Psi_{z^k + \sigma \nabla v^k}(\tilde{z}^k + \sigma \nabla \tilde{v}^k)$$

$$\tilde{x}^{k+1} = (1+\tau)^{-1}\left(\tilde{x}^k + \tau(y + \operatorname{div} \tilde{z}^{k+1})\right)$$

$$\tilde{v}^{k+1} = \tilde{x}^{k+1} + \theta(\tilde{x}^{k+1} - \tilde{x}^k)$$

Differentiation of CP

Π_{B_λ} projection on the λ-ball

Ψ_z hard-thresholding

Example for Anisotropic TV

$$\hat{x}(y) = \underset{x \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{2}\|x - y\|_2^2 + \lambda \|\nabla x\|_1$$

$$
\begin{aligned}
z^{k+1} &= \Pi_{B_\lambda}(z^k + \sigma \nabla v^k) \\
x^{k+1} &= (1+\tau)^{-1}\left(x^k + \tau(y + \operatorname{div} z^{k+1})\right) \\
v^{k+1} &= x^{k+1} + \theta(x^{k+1} - x^k)
\end{aligned}
$$

Chambolle-Pock

$$
\begin{aligned}
\tilde{z}^{k+1} &= \Psi_{z^k + \sigma \nabla v^k}(\tilde{z}^k + \sigma \nabla \tilde{v}^k) \\
\tilde{x}^{k+1} &= (1+\tau)^{-1}\left(\tilde{x}^k + \tau(y + \operatorname{div} \tilde{z}^{k+1})\right) \\
\tilde{v}^{k+1} &= \tilde{x}^{k+1} + \theta(\tilde{x}^{k+1} - \tilde{x}^k)
\end{aligned}
$$

Differentiation of CP

Π_{B_λ} projection on the λ-ball

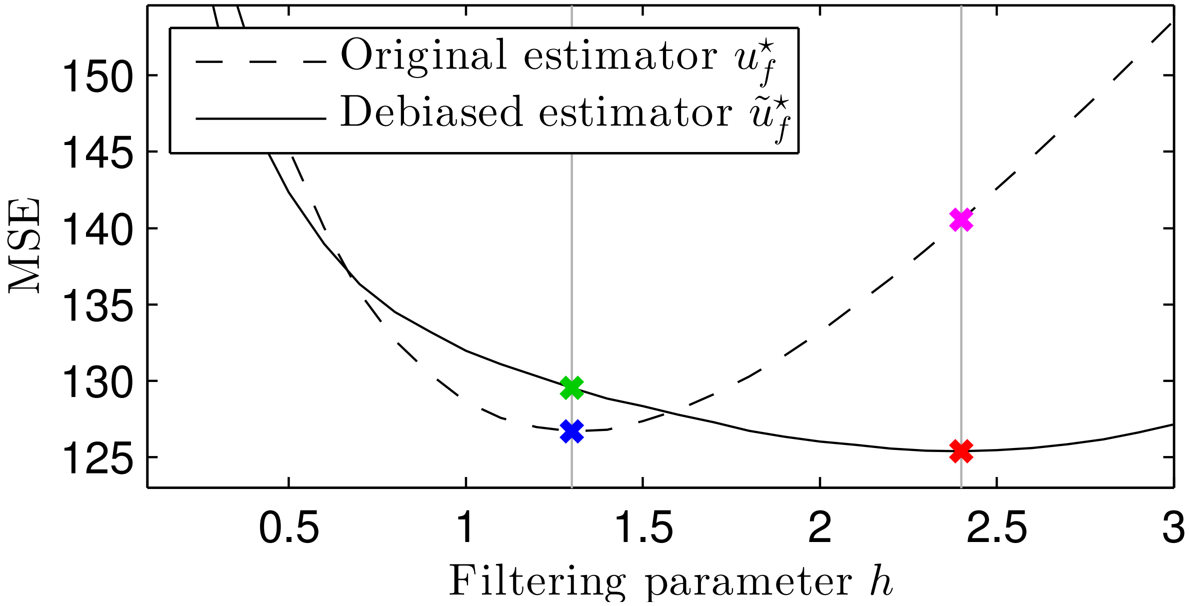
Ψ_z hard-thresholding

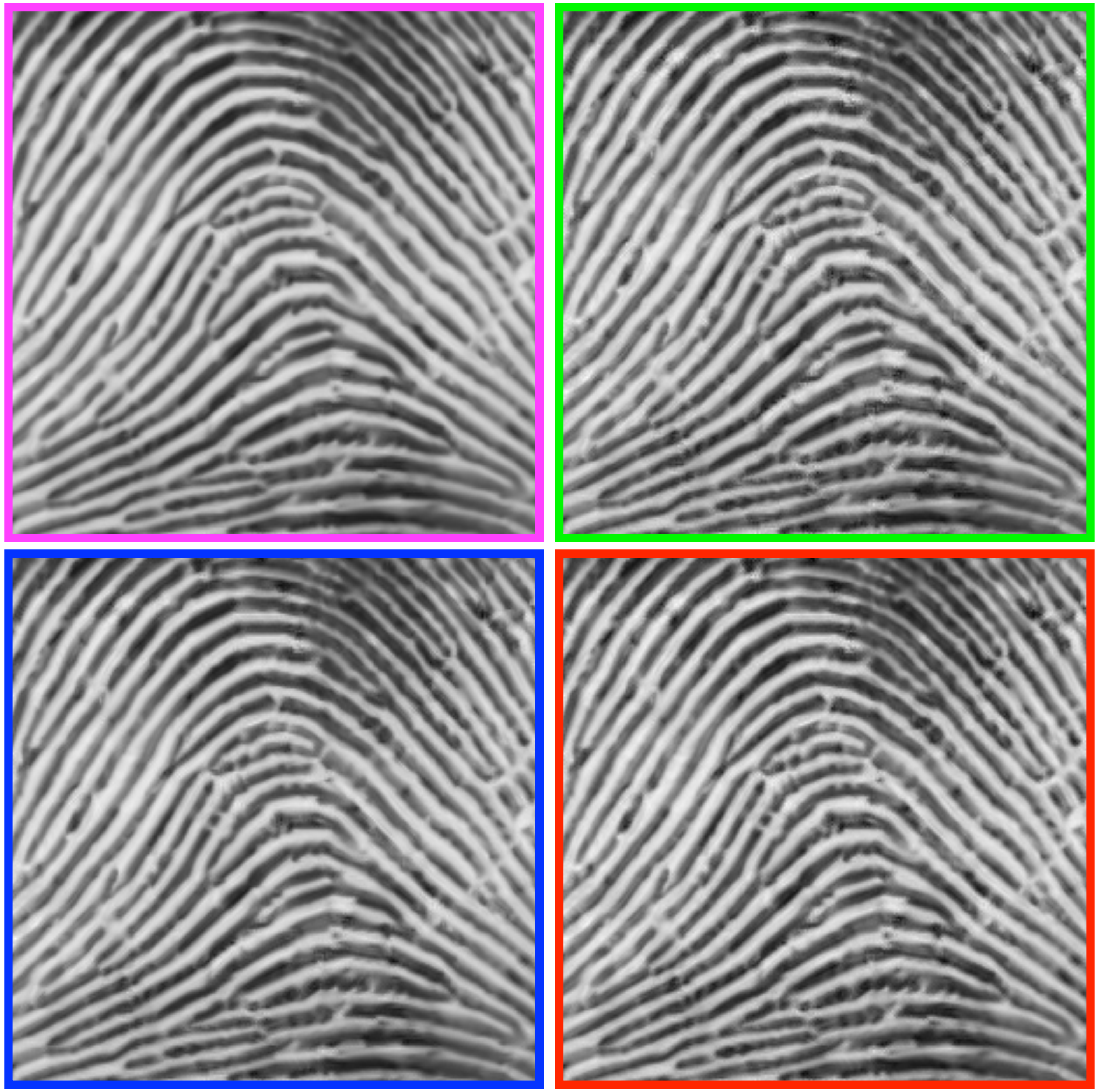
Complexity: 2x CP

Numerical Evaluation for NLM [Buades et al. 2005]

x_0

y



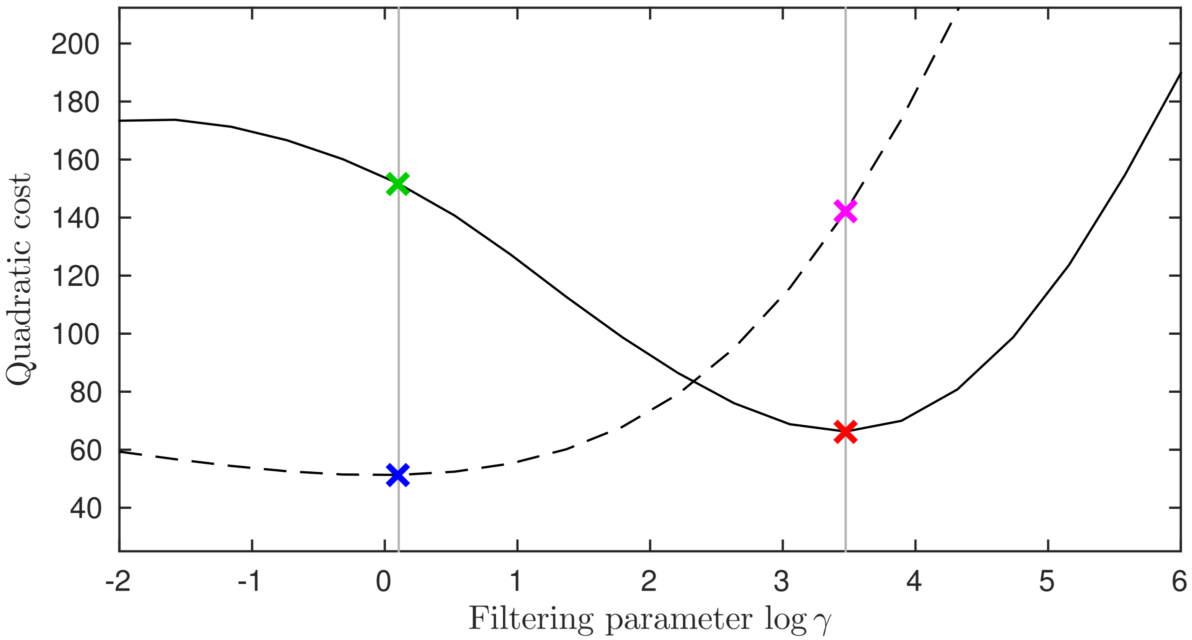


NLM CLEAR

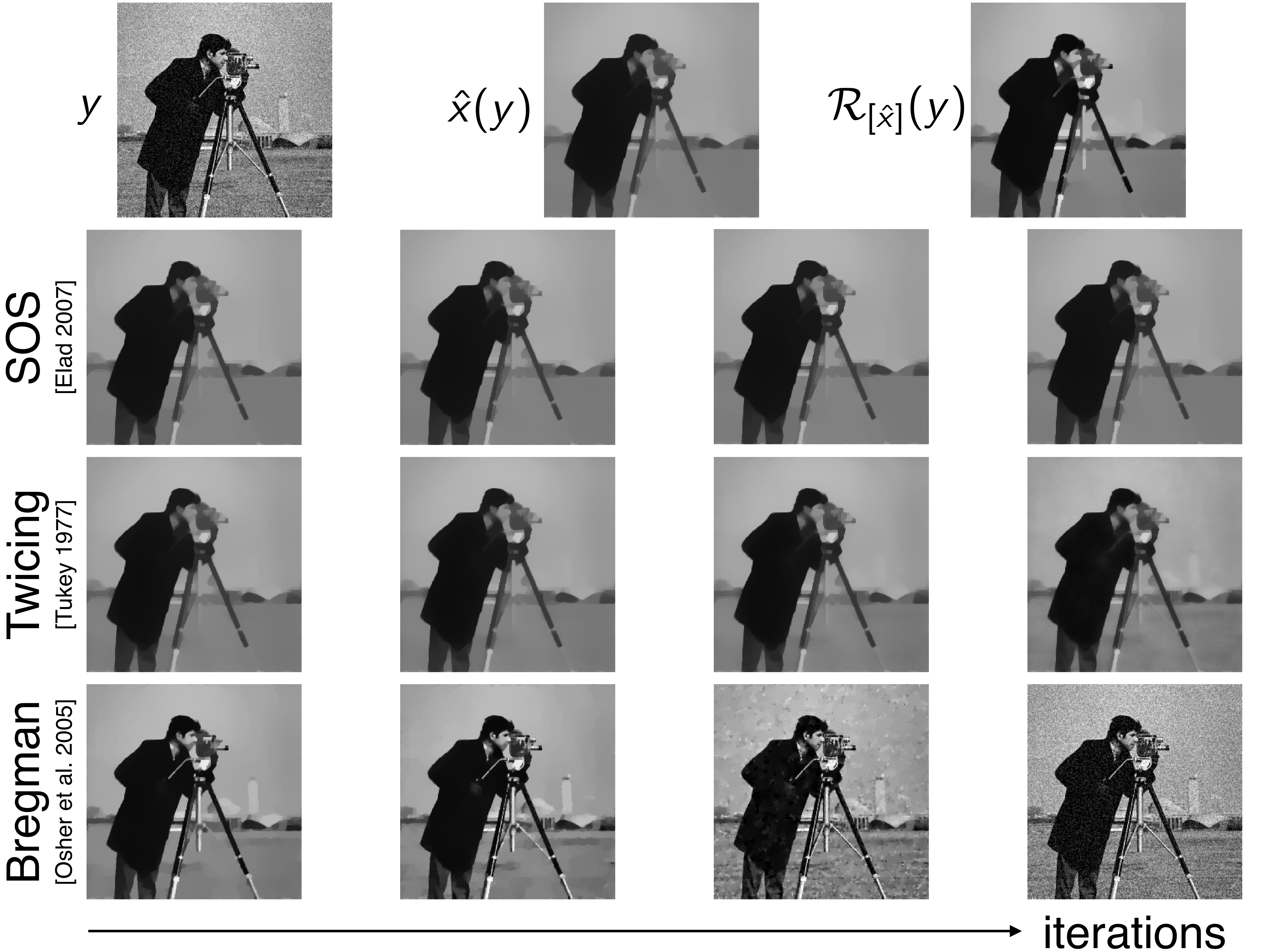
Numerical Evaluation for DDID [Knaus & Zwicker 2013]

x_0

y



DDID CLEAR



y
$\hat{x}(y)$
$\mathcal{R}_{[\hat{x}]}(y)$
SOS
[Elad 2007]
Twicing
[Tukey 1977]
Bregman
[Osher et al. 2005]
iterations

Why not iterating like SOS, Boosting?

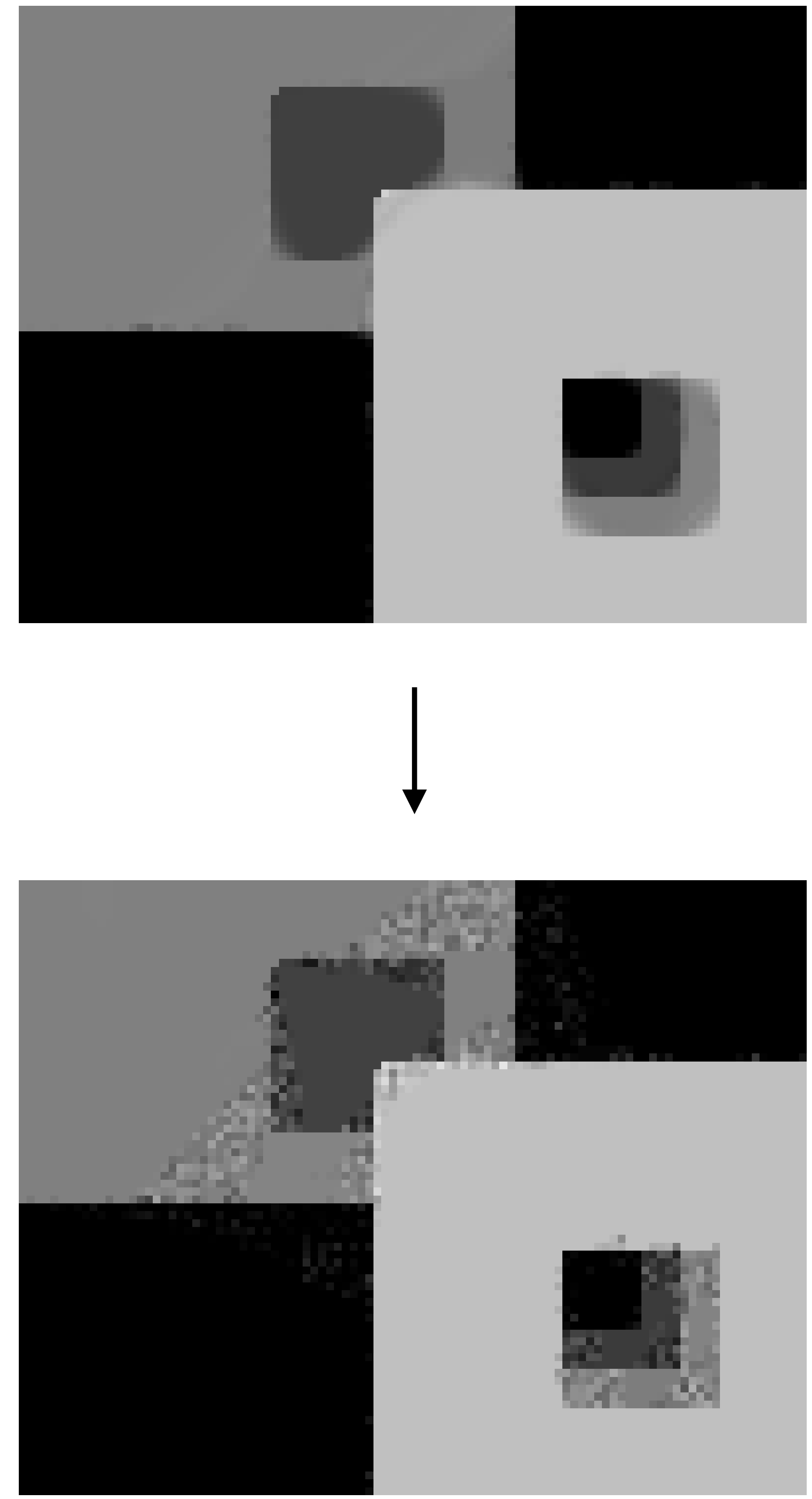
Iterative covariant scheme

$$\left|\begin{array}{l}\tilde{x}_0(y) = \underset{x \in \mathbb{R}^p}{\operatorname{argmin}}\ \frac{1}{2}\|x - y\|_2^2 + \lambda \|\nabla x\|_1 \\ \\ \tilde{x}_{k+1}(y) = \tilde{x}_k(y) + J(y - \tilde{x}_k(y))\end{array}\right.$$

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

∞-covariant $\equiv$ invariant

$\tilde{x}_0$



$\tilde{x}_\infty$

What about CNN?

$$\mathcal{R}_{[\hat{x}]}(y) = \hat{x}(y) + J(y - \hat{x}(y))$$

estimator: **DnCNN** [Zhang et al. 2017] or FFDNet [Zhang et al. 2018]

Noise level 25

Noisy Original DnCNN CLEAR

What about CNN?

$$\mathcal{R}_{[\hat{x}]}(y) = \hat{x}(y) + J(y - \hat{x}(y))$$

estimator: **DnCNN** [Zhang et al. 2017] or FFDNet [Zhang et al. 2018]

Noise level 50

Noisy Original DnCNN CLEAR

What about CNN?

$$\mathcal{R}_{[\hat{x}]}(y) = \hat{x}(y) + J(y - \hat{x}(y))$$

estimator: **DnCNN** [Zhang et al. 2017] or FFDNet [Zhang et al. 2018]

Noise level 150

Noisy | Original | DnCNN | CLEAR

🙁 Residual Network learning noise to remove it

What about CNN?

$$\mathcal{R}_{[\hat{x}]}(y) = \hat{x}(y) + J(y - \hat{x}(y))$$

estimator: DnCNN [Zhang et al. 2017] or **FFDNet** [Zhang et al. 2018]

Noise level 25

Noisy Original FFDNet CLEAR

What about CNN?

$$\mathcal{R}_{[\hat{x}]}(y) = \hat{x}(y) + J(y - \hat{x}(y))$$

estimator: DnCNN [Zhang et al. 2017] or **FFDNet** [Zhang et al. 2018]

Noise level 50

Noisy Original FFDNet CLEAR

What about CNN?

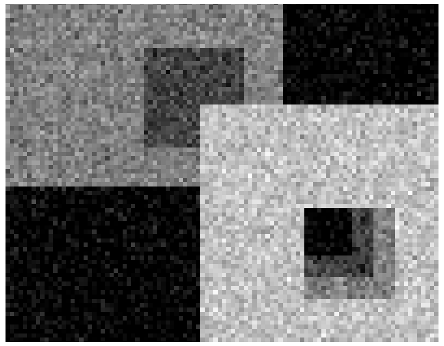
$$\mathcal{R}_{[\hat{x}]}(y) = \hat{x}(y) + J(y - \hat{x}(y))$$

estimator: DnCNN [Zhang et al. 2017] or **FFDNet** [Zhang et al. 2018]

Noise level 150

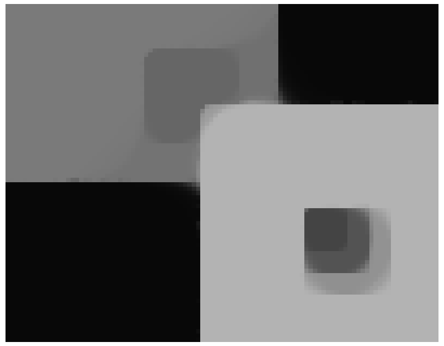
Noisy Original FFDNet CLEAR

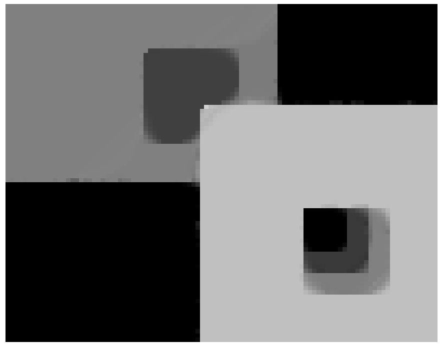
Conclusion



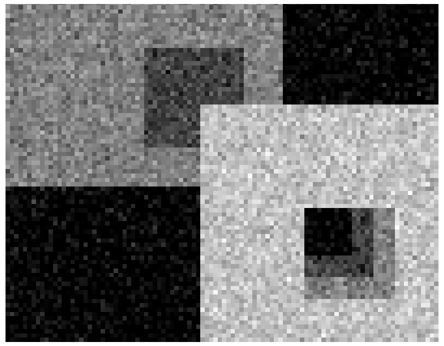
Contributions:

- Fast and accurate denoising re-enhancement
- Parameter-free
- Double the computational cost

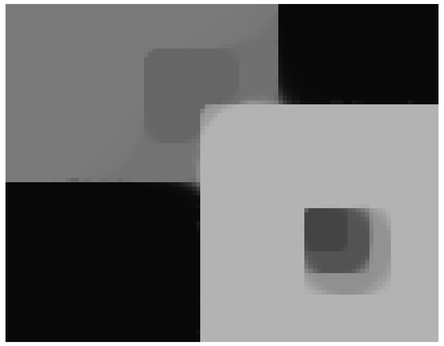




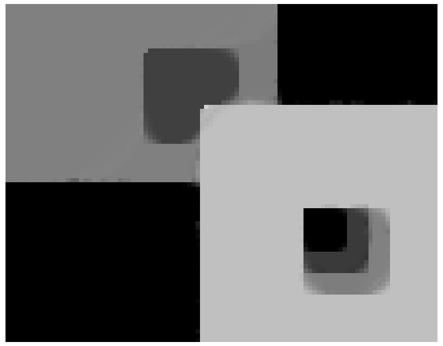
Conclusion



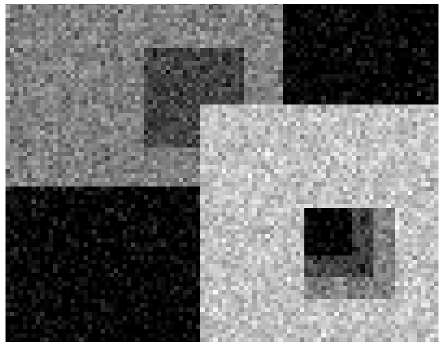
Contributions:
- Fast and accurate denoising re-enhancement
- Parameter-free
- Double the computational cost



Not covered today:
- Bias reduction results [Deledalle et al. 2017]
- Preservation of direction [Brinkman et al. 2016, Deledalle et al. 2019]
- Extension to chrominance [Pierre et al. 2017]

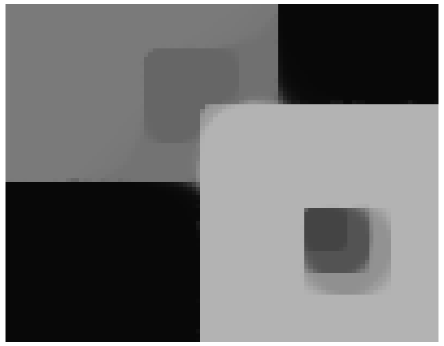


Conclusion



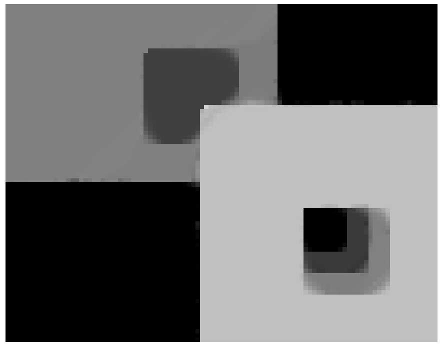
Contributions:

- Fast and accurate denoising re-enhancement
- Parameter-free
- Double the computational cost



Not covered today:

- Bias reduction results [Deledalle et al. 2017]
- Preservation of direction [Brinkman et al. 2016, Deledalle et al. 2019]
- Extension to chrominance [Pierre et al. 2017]



This talk: denoising $y = x_0 + w$

└→ works for inverse problems too $y = \Phi x_0 + w$

(replace J by ΦJ everywhere)

Thanks for your attention!

Some references:

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

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

[Deledalle, Papadakis, Salmon and SV] CLEAR: Covariant LEAst-square Re-fitting. SIAM SIIMS, 2017.

[Deledalle, Papadakis, Salmon and SV] 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.