Multiresolution Block-Coordinate Descent Methods for Imaging Problems

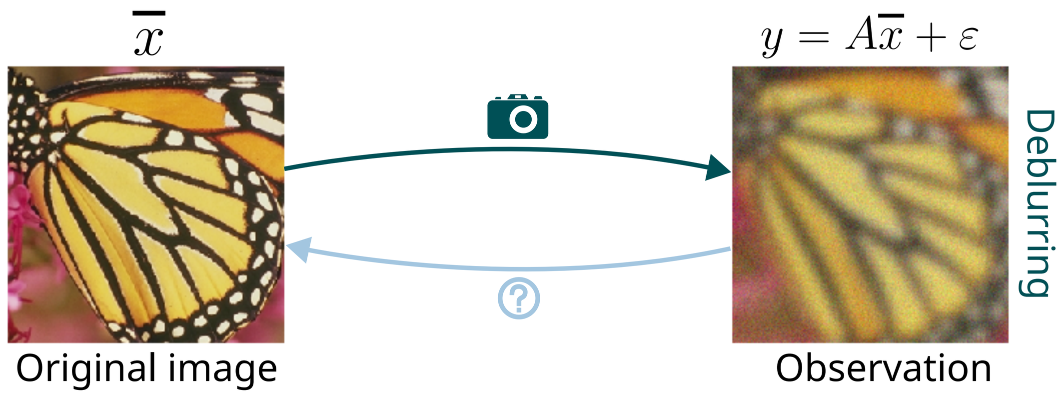
Edgar Desainte-Maréville[1], Marion Foare[1,3], Paulo Gonçalves[1], Nelly Pustelnik[2], Elisa Riccietti[1]

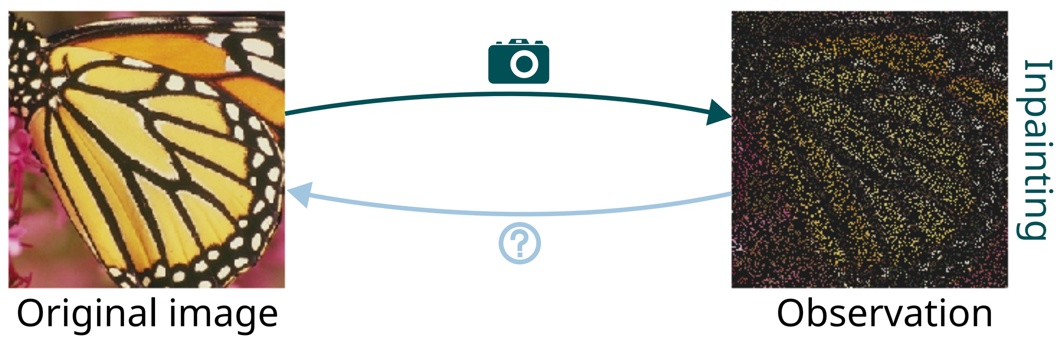
[1]Inria OCKHAM team, LIP, ENS Lyon [2]CNRS, Laboratoire de Physique, ENS Lyon [3]CPE Lyon

Motivation: Study the impact of different update rules in multiresolution block-coordinate descent algorithms.

Imaging inverse problems

Goal: Reconstruct an image from a degraded observation.





Ill-posed inverse problem

The inverse operator may *not exist* or lead to *irregular estimate:*

- Solution may *not be unique*
- Small variations in y ⟶ big variations in $\hat{x}$

Classical optimization method

Build a reconstruction by solving an optimization problem:

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

Data-fidelity: Ensures that $A\hat{x} \approx y$

Regularization: Reduces the search space by imposing some prior on the solution

The **Forward-Backward** algorithm

Given: step size $\tau > 0$, regularization parameter λ
Input: initial point x^0
for $k = 0, 1, 2, \dots$ **do**
$x^{k+\frac{1}{2}} = x^k - \tau A^\top(Ax^k - y)$
$x^{k+1} = \text{prox}_{\tau\lambda R}(x^{k+\frac{1}{2}})$
end for

Reconstruction examples

$R(x) = TV(x) = \|\nabla x\|_1$ $\qquad$ $R(x) = \|Wx\|_1$

Multiresolution decomposition

Wavelet transform: multiscale representation that decomposes an image into its low-frequency part (*approximation coefficients*) and its high-frequency components (*detail coefficients*).

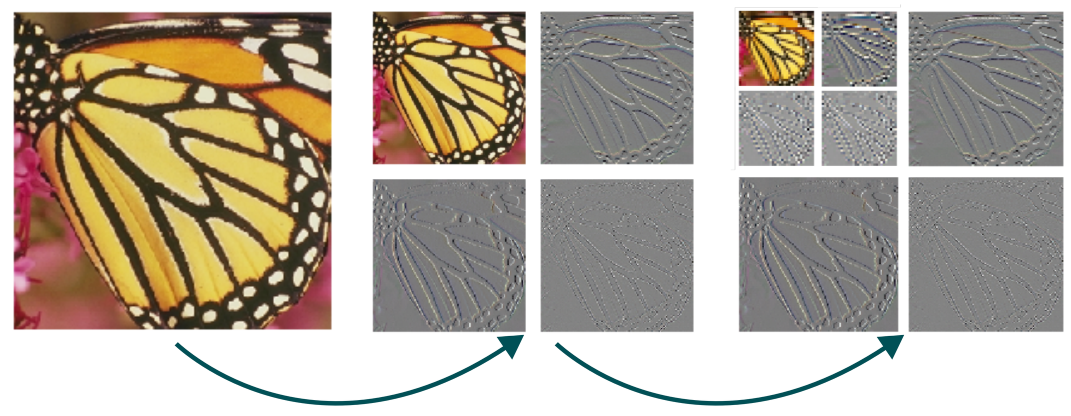
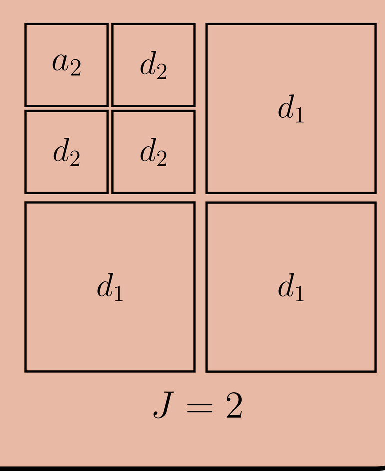


Image as wavelet blocks

Orthonormal wavelet transform: can be performed *recursively* up to a scale J to get a decomposition

$$x = W^\top \underbrace{(a_J, d_J, \dots, d_1)}_{\omega}$$



Block-coordinate descent (BCD) algorithms

Recent advances

Optimization problem written in wavelet domain [1]

$$\hat{x} = W^\top\hat{\omega} \text{ with } \quad \hat{\omega} \in \underset{\omega=(a_J,d_J,\dots,d_1)}{\operatorname{argmin}} \left[\underbrace{\tfrac{1}{2}\|AW^\top(a_J, d_J, \dots, d_1) - y\|_2^2}_{f(a_J,d_J,\dots,d_1)} + \underbrace{\|\Lambda(a_J, d_J, \dots, d_1)\|_1}_{\lambda_{a_J}\|a_J\|_1 + \sum_j \lambda_{d_j}\|d_j\|_1} \right]$$

Block-coordinate descent Forward-Backward algorithm

Given: Step size τ, regularization parameter λ
Input: Initial coefficients $(a_J^0, d_J^0, \dots, d_1^0)$
for $k = 0, 1, 2, \dots$ **do**
$a_J^{k+1} = a_J^k + \varepsilon_{a_J^k}\left(\text{prox}_{\tau\lambda_{a_J}\|\cdot\|_1}(a_J^k - \tau\nabla_{a_J} f(a_J, d_J, \dots, d_1)) - a_J^k\right)$
for $j = J, J-1, \dots, 1$ **do**
$d_j^{k+1} = d_j^k + \varepsilon_{d_j^k}\left(\text{prox}_{\tau\lambda_{d_j}\|\cdot\|_1}(d_j^k - \tau\nabla_{d_j} f(a_J, d_J, \dots, d_1)) - d_J^k\right)$
end for
end for

$(\varepsilon_{a_J^k}, \varepsilon_{d_J^k}, \dots, \varepsilon_{d_1^k}) \in \{0,1\}^{J+1}$ allows to choose which blocks are updated at each iteration

$\varepsilon_{b^k} = 1$: block b is updated at iteration k

$\varepsilon_{b^k} = 0$: block b is not updated at iteration k

Convergence of block-coordinate methods [2]

Under the following assumptions:

- *Convex objective function*
- *Lipschitz-continuous partial gradients*
- *Stepsize bounded by inverse of Lipschitz constant*
- *Every block is updated infinitely many times*

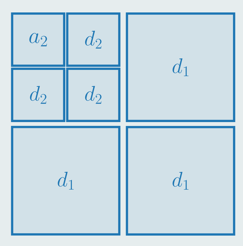
These methods **converge to a minimizer**.

Comparing different update rules

Contributions

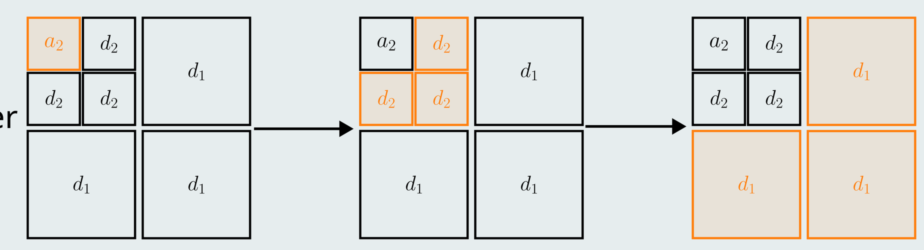
Forward-Backward (FB)

All blocks are updated simultaneously



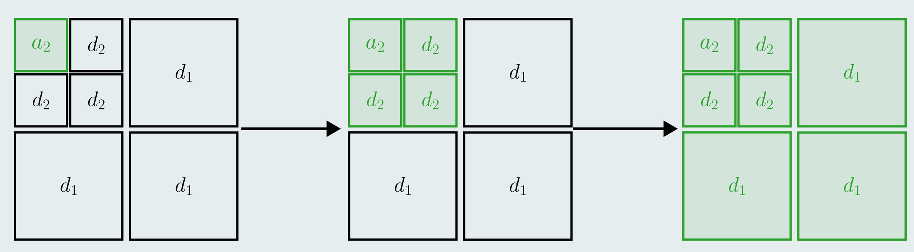
Cyclic

Blocks are updated one after the other from the coarsest scale to the finest



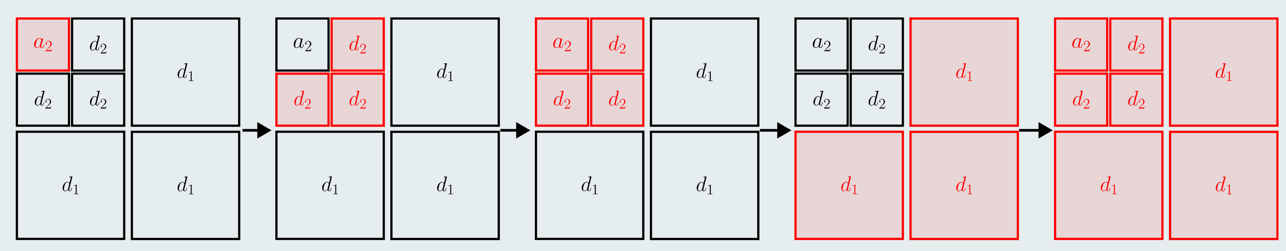
Approximations

Approximation coefficients are updated from the coarsest scale to the finest
(studied in [1])



Approximations then details

Detail coefficients are inserted in between the previous updates



Setting

Ground truth

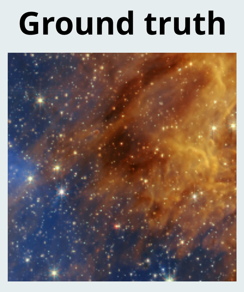


Image size: 1024×1024

Number of **decomposition levels:** 5

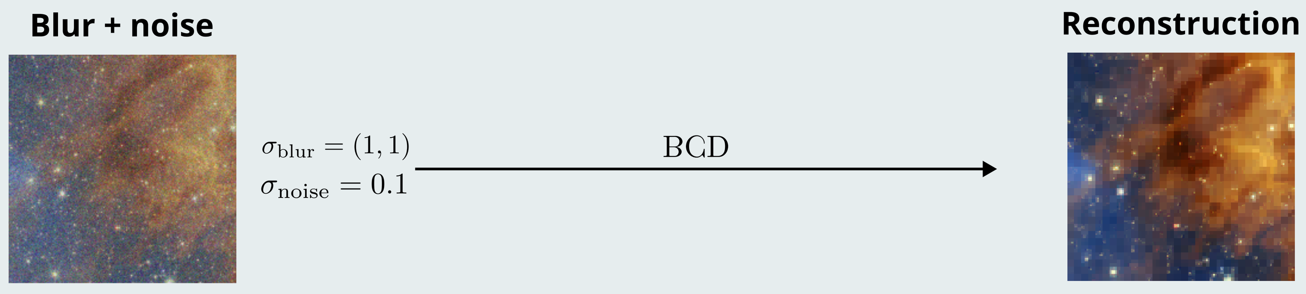
Linear operator A: Gaussian blur with 2D variance σ_{blur}

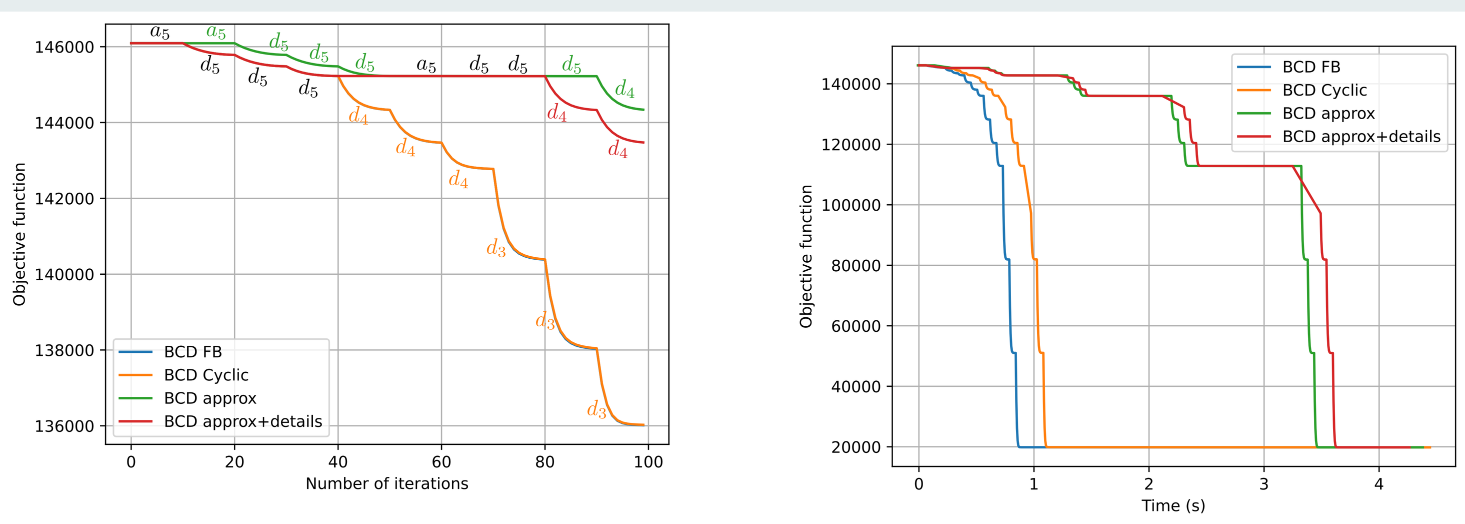
Noise ε: Additive Gaussian noise with variance σ_{noise}

Experiments

Contributions

Low blur / High noise

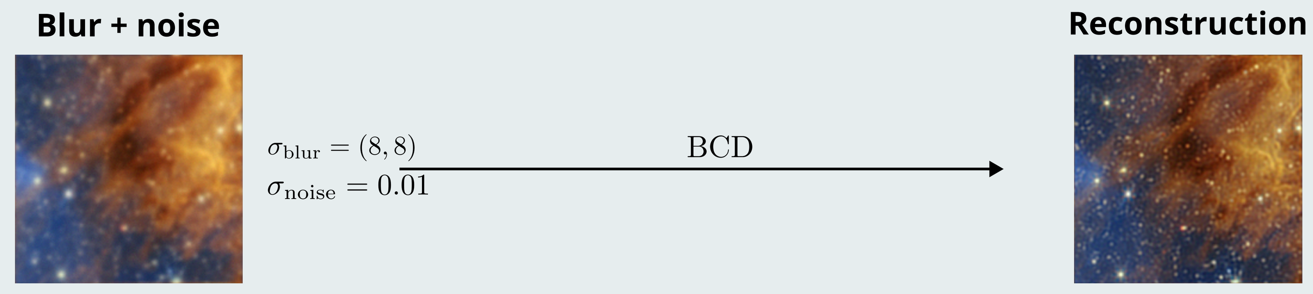


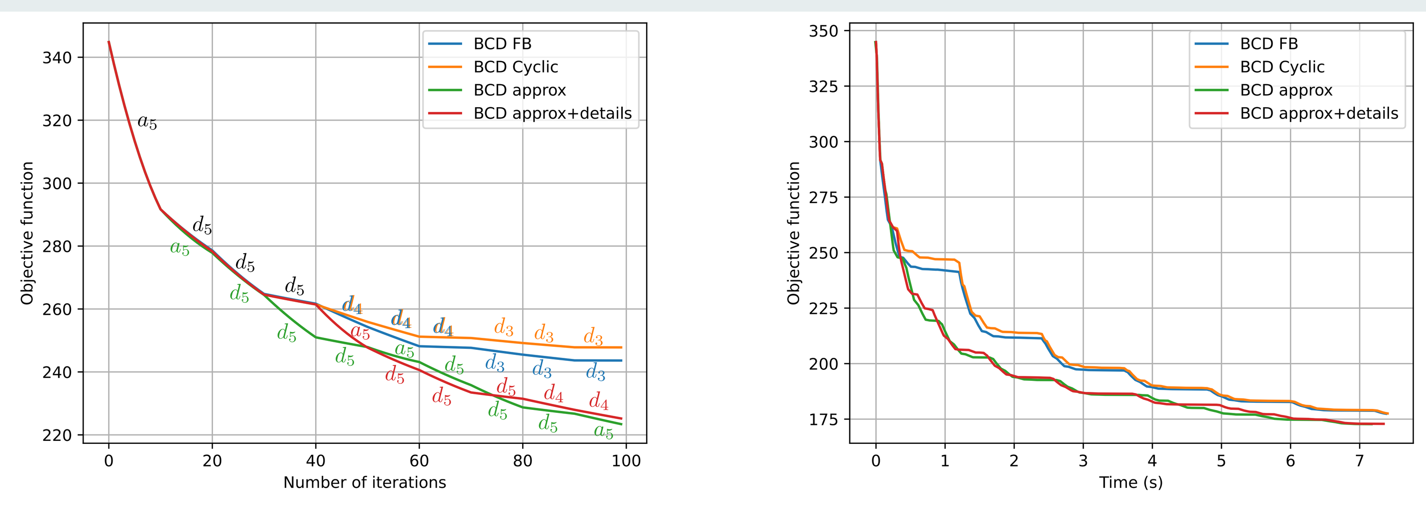


The descent steps on high frequencies are the most efficient

⟶ In a **low blur / high noise** context: favor methods that **focus on high frequencies.**

High blur / Low noise





The descent steps on low frequencies are the most efficient

⟶ In a **high blur / low noise** context: favor methods that **focus on low frequencies.**

[1] Briceño-Arias, L., Gonçalves, P., Lauga, G., Pustelnik, N., Riccietti, E. (2025). *A flexible block-coordinate forward-backward algorithm for non-smooth and non-convex optimization.* arXiv preprint.

[2] Chouzenoux, E., Pesquet, J-C., Repetti, A. (2016). *A block coordinate variable metric forward-backward algorithm.* Journal of Global Optimization, Springer.