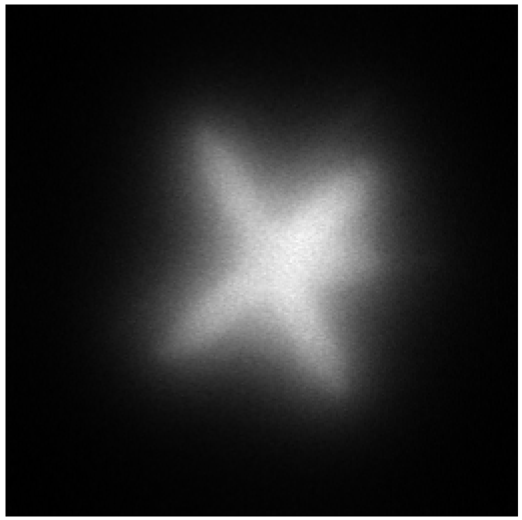
Pietro Dell'Acqua

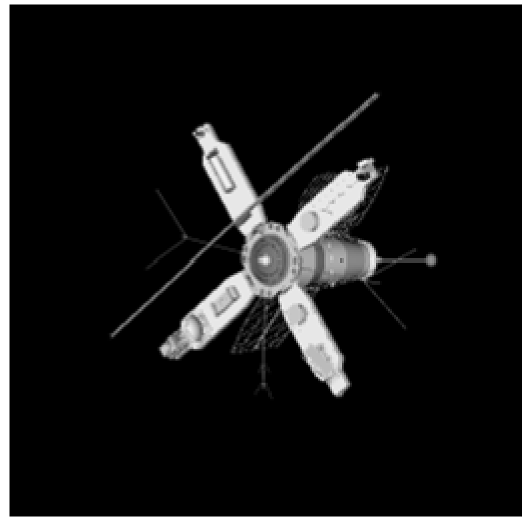
Fast and accurate numerical techniques for deblurring models

PING Workshop
Opening Meeting for the Research Project GNCS 2016
"PING - Inverse Problems in Geophysics"
Florence, April 6, 2016

The image restoration problem



Recorded image



True image

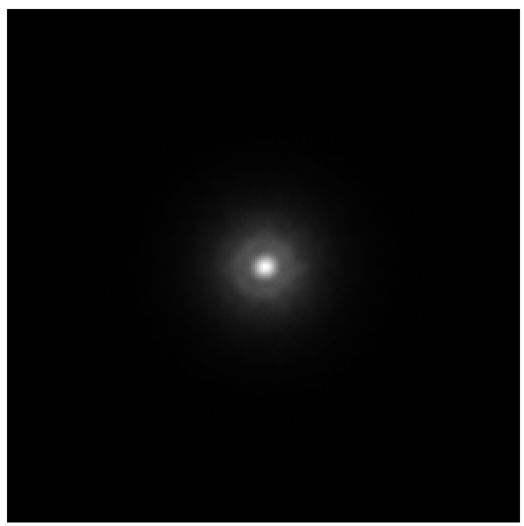
By the knowledge of the observed data (*the effect*), we want to find an approximation of the true image (*the cause*).

Blurring model

Classical image deblurring problem with space invariant blurring. Under such assumption the image formation process is modelled by

$$b(s) = \int h(s-t)\bar{x}(t)dt + \eta(s), \;\; s \in \mathbb{R}^2$$

where h is the known impulse response of the imaging system, i.e. the point spread function (PSF), $\bar{x}$ denotes the true physical object, η takes into account measurement errors and noise.



Point spread function

Discrete problem

We have to solve the linear equation

$$Ax = b,$$

where A is the blurring matrix and $b = A\bar{x} + \eta$ is the blurred and noisy image.

The associated system of normal equations

$$A^H Ax = A^H b$$

is solved in order to find an approximated least squares solution.

A is a large ill-conditioned matrix

A (h_{PSF}, BCs) is a structured matrix

Structured matrices

- Zero BCs: Block Toeplitz with Toeplitz blocks (BTTB)
- Periodic BCs: Block circulant with circulant blocks (BCCB)
  - FFT (Fast Fourier Transform)
- Reflective BCs: Block Toeplitz+Hankel with Toeplitz+Hankel blocks
  - DCT (Discrete Cosine Transform) { for symmetric PSFs }
- Anti-Reflective BCs: Block Toeplitz+Hankel with Toeplitz+Hankel blocks + a low rank matrix
  - ART (Anti-Reflective Transform) { for symmetric PSFs }

Research activity

[1] P. Dell'Acqua, M. Donatelli, S. Serra Capizzano, D. Sesana, C. Tablino Possio, *Optimal preconditioning for image deblurring with Anti-Reflective boundary conditions*, Linear Algebra and its Applications, in press (2015).

[2] P. Dell'Acqua, M. Donatelli, C. Estatico, *Preconditioners for image restoration by reblurring techniques*, Journal of Computational and Applied Mathematics **272**, pp. 313–333 (2014).

[3] P. Dell'Acqua, M. Donatelli, C. Estatico, M. Mazza, *Structure preserving preconditioners for image deblurring*, submitted.

[4] P. Dell'Acqua, C. Estatico, *Acceleration of multiplicative iterative algorithms for image deblurring by duality maps in Banach spaces*, Applied Numerical Mathematics **99**, 121–136 (2016).

[5] P. Dell'Acqua, *ν acceleration of statistical iterative methods for image restoration*, Signal, Image and Video Processing, in press.

Optimal preconditioning

Z variant

Acceleration techniques

Optimal preconditioning

Let $A = A(h)$ be the Anti-Reflective matrix generated by the generic PSF $h_{PSF} = \left[h_{i_1,i_2}\right]_{i_1=-q_1,\ldots,q_1, i_2=-q_2,\ldots,q_2}$ and let $P = P(s) \in \mathcal{AR}_n^{2D}$ be the Anti-Reflective matrix generated by the symmetrized PSF $s_{PSF} = \left[s_{i_1,i_2}\right]_{i_1=-q_1,\ldots,q_1, i_2=-q_2,\ldots,q_2}$.

We are looking for the optimal preconditioner $P^* = P^*(s^*)$ in the sense that

$$P^* = \arg \min_{P \in \mathcal{AR}_n^{2D}} \|A - P\|_{\mathcal{F}}^2 \;, \quad s^* = \arg_s \min \|A(h) - P(s)\|_{\mathcal{F}}^2 \;,$$

where $\|\cdot\|_{\mathcal{F}}$ is the Frobenius norm, defined as $\|A\|_{\mathcal{F}} = \sqrt{\sum_{i,j} \left|a_{i,j}\right|^2}$.

Optimal preconditioning

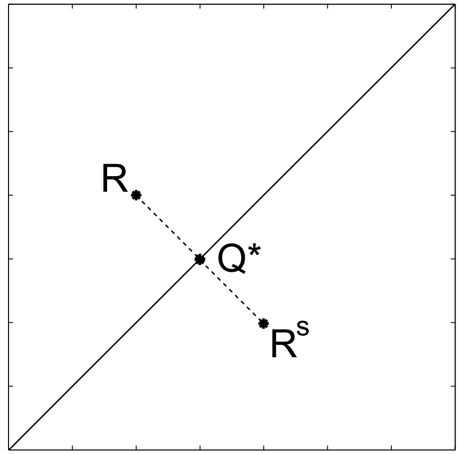
The result is known for Reflective BCs.

Given a generic PSF h_{PSF}, the optimal preconditioner in the DCT matrix algebra is generated by the strongly symmetric PSF s_{PSF} given by

$$\begin{aligned} 1\mathrm{D}: &\;\; s_{\pm i} = \frac{h_{-i} + h_i}{2}, \\ 2\mathrm{D}: &\;\; s_{\pm i_1, \pm i_2} = \frac{h_{-i_1,-i_2} + h_{-i_1,i_2} + h_{i_1,-i_2} + h_{i_1,i_2}}{4}, \end{aligned}$$

which is a symmetrization of the original PSF.

Geometrical idea of the proof - 1D



A point R, its swapped point R^S, the optimal approximation of both Q^*.

We simply observe that if we consider in the Cartesian plane a point $R = (R_x, R_y)$, its optimal approximation Q^*, among the points $Q = (Q_x, Q_y)$ such that $Q_x = Q_y$, is obtained as the intersection between the line $y = x$ with the perpendicular line that pass through R, that is

$$\begin{cases} y - R_y = -(x - R_x) \\ y = x \end{cases}$$

hence $Q_x^* = Q_y^* = (R_x + R_y)/2$. The same holds true if we consider the swapped point $R^S = (R_y, R_x)$, since they share the same distance, i.e. $d(R, Q^*) = d(R^S, Q^*)$. Clearly, due to linearity of obtained expression, this result can be extended also to the case of any linear combination of coordinates.

Geometrical idea of the proof - 1D

For the sake of simplicity we report a small example

$$A-P=\begin{bmatrix} \omega_0^y & \nu_1^x & \nu_2^x & \nu_3^x & & & \\ \omega_1^y & \zeta_0^y & \zeta_2^x & \vartheta_2^x & \vartheta_3^x & & \\ \omega_2^y & \zeta_1^y & \vartheta_0 & \vartheta_1^x & \vartheta_2^x & \vartheta_3^x & \\ \omega_3^y & \vartheta_2^y & \vartheta_1^y & \vartheta_0 & \vartheta_1^x & \vartheta_2^x & \omega_3^x \\ & \vartheta_3^y & \vartheta_2^y & \vartheta_1^y & \vartheta_0 & \zeta_1^x & \omega_2^x \\ & & \vartheta_3^y & \vartheta_2^y & \zeta_2^y & \zeta_0^x & \omega_1^x \\ & & & \nu_3^y & \nu_2^y & \nu_1^y & \omega_0^x \end{bmatrix} - \begin{bmatrix} \hat\omega_0^y & 0 & 0 & 0 & & & \\ \hat\omega_1^y & \hat\zeta_0^y & \hat\zeta_2^x & \hat\vartheta_2^x & \hat\vartheta_3^x & & \\ \hat\omega_2^y & \hat\zeta_1^y & \hat\vartheta_0 & \hat\vartheta_1^x & \hat\vartheta_2^x & \hat\vartheta_3^x & \\ \hat\omega_3^y & \hat\vartheta_2^y & \hat\vartheta_1^y & \hat\vartheta_0 & \hat\vartheta_1^x & \hat\vartheta_2^x & \hat\omega_3^x \\ & \hat\vartheta_3^y & \hat\vartheta_2^y & \hat\vartheta_1^y & \hat\vartheta_0 & \hat\zeta_1^x & \hat\omega_2^x \\ & & \hat\vartheta_3^y & \hat\vartheta_2^y & \hat\zeta_2^y & \hat\zeta_0^x & \hat\omega_1^x \\ & & & 0 & 0 & 0 & \hat\omega_0^x \end{bmatrix}$$

Here, we set the points

$$\begin{aligned} \Theta_i &= (\vartheta_i^x, \vartheta_i^y) = (h_{-i}, h_i) \\ \Omega_i &= (\omega_i^x, \omega_i^y) = (\vartheta_i^x + 2\sum_{j=i+1}^{q} \vartheta_j^x, \vartheta_i^y + 2\sum_{j=i+1}^{q} \vartheta_j^y) \\ N_i &= (\nu_i^x, \nu_i^y) = (h_{-i} - h_i, h_i - h_{-i}) = (\vartheta_i^x - \vartheta_i^{Sx}, \vartheta_i^y - \vartheta_i^{Sy}) \\ Z_0 &= (\zeta_0^x, \zeta_0^y) = (h_0 - h_{-2}, h_0 - h_2) = (\vartheta_0^x - \vartheta_2^x, \vartheta_0^y - \vartheta_2^y) \\ Z_1 &= (\zeta_1^x, \zeta_1^y) = (h_{-1} - h_{-3}, h_1 - h_3) = (\vartheta_1^x - \vartheta_3^x, \vartheta_1^y - \vartheta_3^y) \\ Z_2 &= (\zeta_2^x, \zeta_2^y) = (h_{-1} - h_3, h_1 - h_{-3}) = (\vartheta_1^x - \vartheta_3^{Sx}, \vartheta_1^y - \vartheta_3^{Sy}) \end{aligned}$$

Geometrical idea of the proof - 2D

We simply observe that if we consider in the 4-dimensional space a point $R = (R_x, R_y, R_z, R_w)$, its optimal approximation Q^* among the points $Q = (Q_x, Q_y, Q_z, Q_w)$ belonging to the line $\mathcal{L}$

$$\begin{cases} x = t \\ y = t \\ z = t \\ w = t \end{cases}$$

is obtained by minimizing the distance

$$\begin{aligned} \mathbf{d}^2(\mathcal{L}, R) &= (t - R_x)^2 + (t - R_y)^2 + (t - R_z)^2 + (t - R_w)^2 \\ &= 4t^2 - 2t(R_x + R_y + R_z + R_w) + R_x^2 + R_y^2 + R_z^2 + R_w^2. \end{aligned}$$

This is a trinomial of the form $\alpha t^2 + \beta t + \gamma$, with $\alpha > 0$ and we find the minimum by using the formula for computing the abscissa of the vertex of a parabola

$$t^* = -\frac{\beta}{2\alpha} = \frac{R_x + R_y + R_z + R_w}{4}.$$

Hence the point Q^* is defined as $Q_x^* = Q_y^* = Q_z^* = Q_w^* = t^*$. The same holds true if we consider any swapped point R^S, not unique but depending on the permutation at hand, since they share the same distance, i.e. $d(R, Q^*) = d(R^S, Q^*)$. Again, thanks to the linearity of obtained expression, this result can be extended also in the case of any linear combination of coordinates.

Iterative regularization methods

Van Cittert method

$x_k = x_{k-1} + \tau(b - Ax_{k-1})$

Landweber method

$x_k = x_{k-1} + \tau A^H(b - Ax_{k-1})$

Steepest descent method

$x_k = x_{k-1} + \tau_{k-1} A^H(b - Ax_{k-1})$

$\tau_{k-1} = \|r_{k-1}\|_2^2 / \|Ar_{k-1}\|_2^2$, with $r_{k-1} = A^H(b - Ax_{k-1})$

Lucy-Richardson method (LR)

$x_k = x_{k-1} \cdot A^H \left(\frac{b}{Ax_{k-1}}\right)$

Image Space Reconstruction Algorithm (ISRA)

$x_k = x_{k-1} \cdot \left(\frac{A^H b}{A^H A x_{k-1}}\right)$

The idea

All the algorithms presented base the update of the iteration on the "key" quantities

$$b - Ax_{k-1} \quad \text{or} \quad \frac{b}{Ax_{k-1}},$$

which both give information on the distance between the blurred data b and the blurred iteration Ax_{k-1}.

A^H can be seen as a *reblurring* operator, whose role is basically to help the method to manage the noise.

Our idea is to pick a new matrix Z, which will replace A^H.

We notice that in principle one can think to choose Z as another operator, not necessarily related to a blurring process.

Z variant

Z-Landweber method

$$x_k = x_{k-1} + \tau Z(b - Ax_{k-1})$$

Z-Steepest descent method

$$x_k = x_{k-1} + \tau_{k-1} Z(b - Ax_{k-1})$$

$$\tau_{k-1} = \frac{r_{k-1}^H r_{k-1}}{r_{k-1}^H Z A r_{k-1}}, \text{ with } r_{k-1} = Z(b - Ax_{k-1})$$

Z-LR

$$x_k = x_{k-1} \cdot Z\left(\frac{b}{Ax_{k-1}}\right)$$

Z-ISRA

$$x_k = x_{k-1} \cdot \left(\frac{Zb}{ZAx_{k-1}}\right)$$

Link with preconditioning

The conventional preconditioned system is the following

$$DA^H Ax = DA^H b,$$

where D is the preconditioner, whose role is to suitably approximate the (generalized) inverse of the normal matrix $A^H A$.

The new strategy leads to the new preconditioned system

$$Z Ax = Z b\,,$$

whose aim is to allow iterative methods to become faster and more stable.

- p Low Pass Filter

$$d_j = \begin{cases} 0 & \text{if } |\lambda_j| < \zeta \\ 1/|\lambda_j|^p & \text{if } |\lambda_j| \geq \zeta \end{cases}$$

- p Hanke Nagy Plemmons Filter

$$d_j = \begin{cases} 1 & \text{if } |\lambda_j| < \zeta \\ 1/|\lambda_j|^p & \text{if } |\lambda_j| \geq \zeta \end{cases}$$

- p Tyrtyshnikov Yeremin Zamarashkin Filter

$$d_j = \begin{cases} 1/\zeta & \text{if } |\lambda_j| < \zeta \\ 1/|\lambda_j|^p & \text{if } |\lambda_j| \geq \zeta \end{cases}$$

- Tikhonov Filter

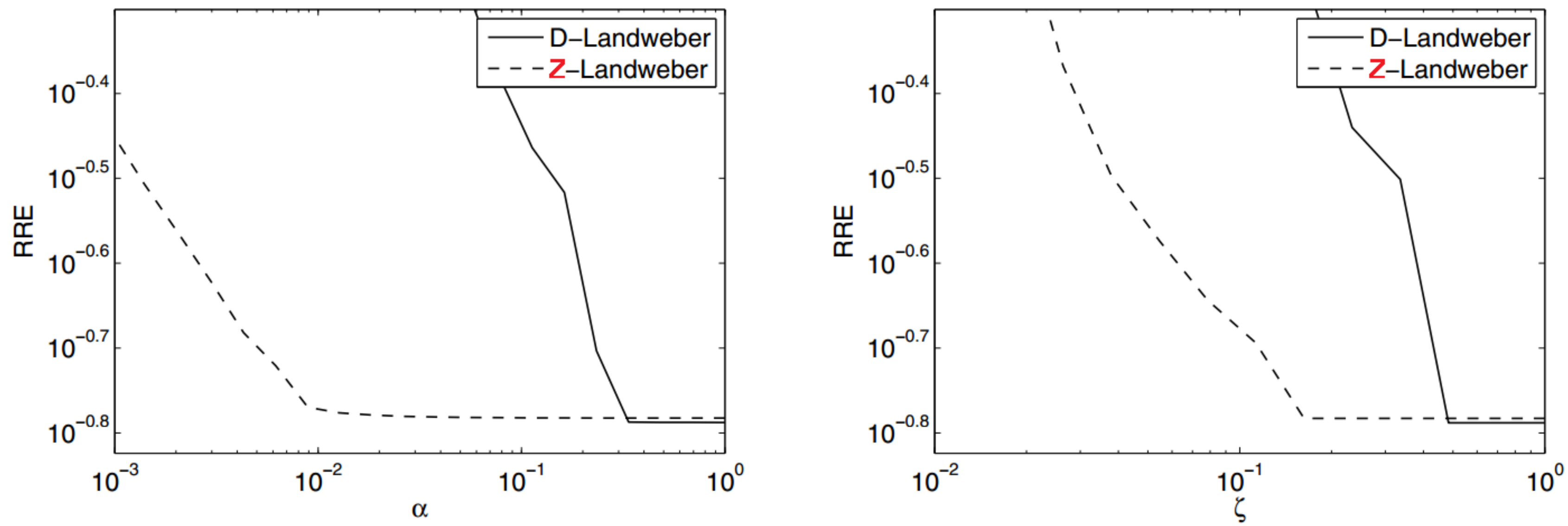
$$d_j = \frac{1}{|\lambda_j|^2 + \alpha}$$

By using each filter we can define the eigenvalues of Z as

$$z_j = \bar{\lambda}_j d_j$$

BCCB preconditioning: D vs Z

Reflective and Anti-Reflective BCs



RRE vs regularization parameter for Tikhonov filter (α) and T.Y.Z. filter (ζ).

For all filters Z variant shows an higher *stability*, and with this word we mean that iterative methods compute a good restoration also when regularization parameters ζ and α are very small.

A general Z algorithm

Called c_j the eigenvalues of the BCCB matrix associated with (h_{PSF}, 'periodic'), for any BCs, we can perform the next algorithm.

$Z \longleftarrow$ ALGORITHM(h_{PSF}, BCs)

- get $\{c_j\}_{j=1}^{n^2}$ by computing FFT of h_{PSF}
- get z_j by applying a filter to c_j
- get w_{PSF} by computing IFFT of $\{z_j\}_{j=1}^{n^2}$
- generate Z from (w_{PSF}, BCs)

The algorithm is consistent, in fact if the filter is identity, i.e. there is no filtering, we have $Z = A^H$. Clearly an analogous algorithm can be applied to create the preconditioner D.

ν acceleration

The so-called ν-method is defined as follows

$$x_k = \mu_k x_{k-1} + (1 - \mu_k) x_{k-2} + \omega_k A^H (b - A x_{k-1}),$$

where the coefficients μ_k and ω_k are given by

$$\begin{aligned} \mu_k &= 1 + \frac{(k-1)(2k-3)(2k+2\nu-1)}{(k+2\nu-1)(2k+4\nu-1)(2k+2\nu-3)}, \\ \omega_k &= \frac{4(2k+2\nu-1)(k+\nu-1)}{(k+2\nu-1)(2k+4\nu-1)}, \end{aligned}$$

for $k > 1$, and with $\mu_1 = 1$, $\omega_1 = 1$.

ν acceleration

We rewrite LR in this way

$$x_k = x_{k-1} + \left[x_{k-1} \cdot A^H \left(\frac{b}{Ax_{k-1}} \right) - x_{k-1} \right],$$

whence we have

$$\begin{aligned} x_k &= \mu_k x_{k-1} + (1-\mu_k) x_{k-2} + \omega_k \left[x_{k-1} \cdot A^H \left(\frac{b}{Ax_{k-1}} \right) - x_{k-1} \right] \\ &= (\mu_k - \omega_k) x_{k-1} + (1-\mu_k) x_{k-2} + \omega_k \left[x_{k-1} \cdot A^H \left(\frac{b}{Ax_{k-1}} \right) \right]. \end{aligned}$$

An analogous formula holds for ISRA

$$x_k = (\mu_k - \omega_k) x_{k-1} + (1-\mu_k) x_{k-2} + \omega_k \left[x_{k-1} \cdot \left(\frac{A^H b}{A^H A x_{k-1}} \right) \right].$$

Automatic acceleration

The most popular acceleration technique, introduced in 1997 by Biggs and Andrews.

It is a form of vector extrapolation that predicts subsequent points based on previous points.

$$
\begin{aligned}
y_k &= x_k + \alpha_k (x_k - x_{k-1}), \\
\alpha_k &= \frac{(g_{k-1})^T g_{k-2}}{(g_{k-2})^T g_{k-2}}, \\
g_{k-1} &= x_k - y_{k-1}, \\
g_{k-2} &= x_{k-1} - y_{k-2}, \\
x_{k+1} &= \text{it.method}(y_k),
\end{aligned}
$$

where $\alpha_1 = 0$, $\alpha_2 = 0$, $0 \leq \alpha_k \leq 1$, $\forall k$.

Numerical results

Table 1 Satellite test: best RRE with relative IT and computation time of classical LR and ISRA

	Time (s)	RRE	IT
LR	266.1	0.3484	2128
ISRA	242.2	0.3451	1926

Table 2 Cameraman test: best RRE with relative IT and computation time of classical LR and ISRA

	Time (s)	RRE	IT
LR	57.6	0.0798	484
ISRA	74.5	0.0770	605

Table 3 Satellite test: best RRE with relative IT, AF and computation time for accelerated methods

		Aut acc				ν acc											
						$\nu = 0.7$				$\nu = 1$				$\nu = 2$			
		Time (s)	RRE	IT	AF	Time (s)	RRE	IT	AF	Time (s)	RRE	IT	AF	Time (s)	RRE	IT	AF
LR	$\mathcal{P}$	11.2	0.3646	83	10.1	5.9	0.3650	51	16.4	8.6	0.3526	69	20.4	11.4	0.3472	95	24.7
	$\mathcal{Q}$	15.2	0.3555	123	9.9	6.3	0.3601	55	18.2	8.6	0.3478	70	32.2	11.4	0.3471	95	25.0
ISRA	$\mathcal{P}$	9.3	0.3695	75	7.5	2.9	0.3840	24	15.0	7.9	0.3484	62	21.1	10.5	0.3443	89	23.8
	$\mathcal{Q}$	13.0	0.3560	109	8.4	5.8	0.3539	47	21.1	7.5	0.3458	65	24.9	10.9	0.3442	89	23.8

Table 4 Cameraman test: best RRE with relative IT, AF and computation time for accelerated methods

		Aut acc				ν acc											
						$\nu = 0.7$				$\nu = 1$				$\nu = 2$			
		Time (s)	RRE	IT	AF	Time (s)	RRE	IT	AF	Time (s)	RRE	IT	AF	Time (s)	RRE	IT	AF
LR	$\mathcal{P}$	5.1	0.0836	41	5.2	3.3	0.0828	27	8.7	3.9	0.0814	31	9.3	5.0	0.0804	42	8.5
	$\mathcal{Q}$	5.8	0.0826	48	5.1	3.3	0.0824	27	9.3	3.9	0.0813	31	9.5	5.0	0.0804	42	8.5
ISRA	$\mathcal{P}$	5.9	0.0811	47	5.8	3.5	0.0802	29	10.4	4.5	0.0786	34	11.0	6.0	0.0775	48	9.8
	$\mathcal{Q}$	7.1	0.0791	58	6.0	3.6	0.0785	30	12.7	4.1	0.0779	35	12.1	6.0	0.0774	48	10.0

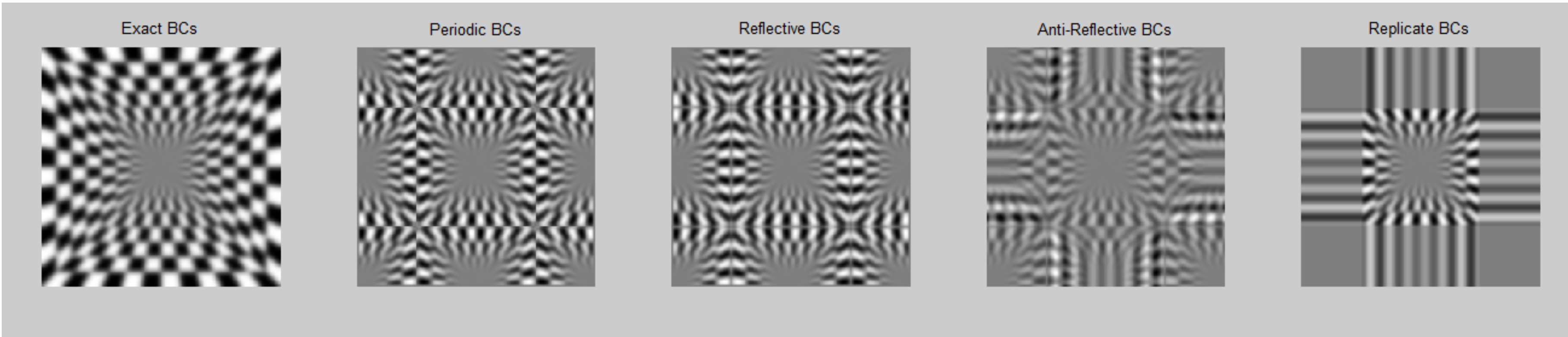
An alternative approach

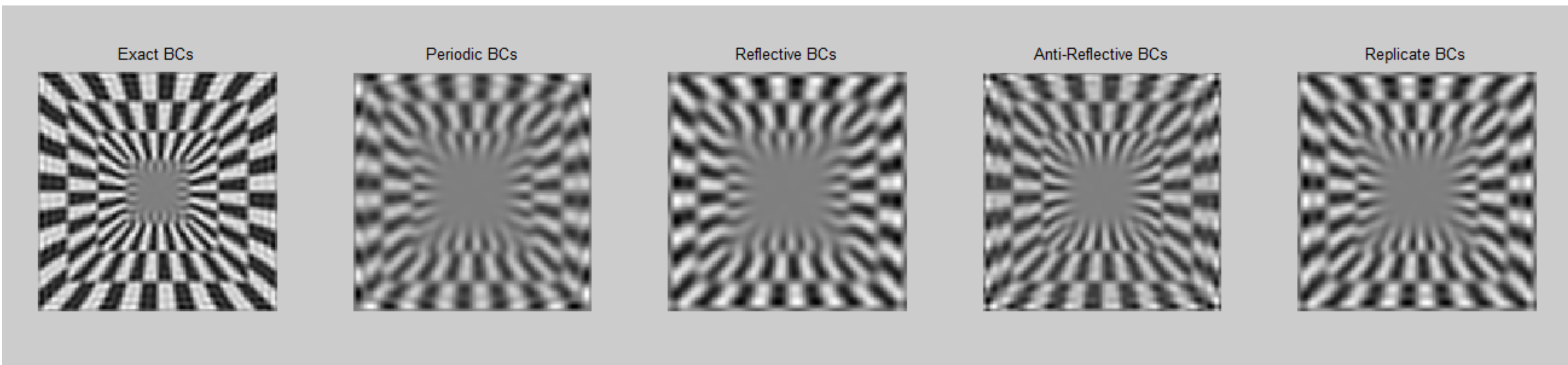
Instead of considering A as a structured matrix (whose structure depends on BCs), an alternative approach consists in solving

$$Cx_{ext} = b_{ext},$$

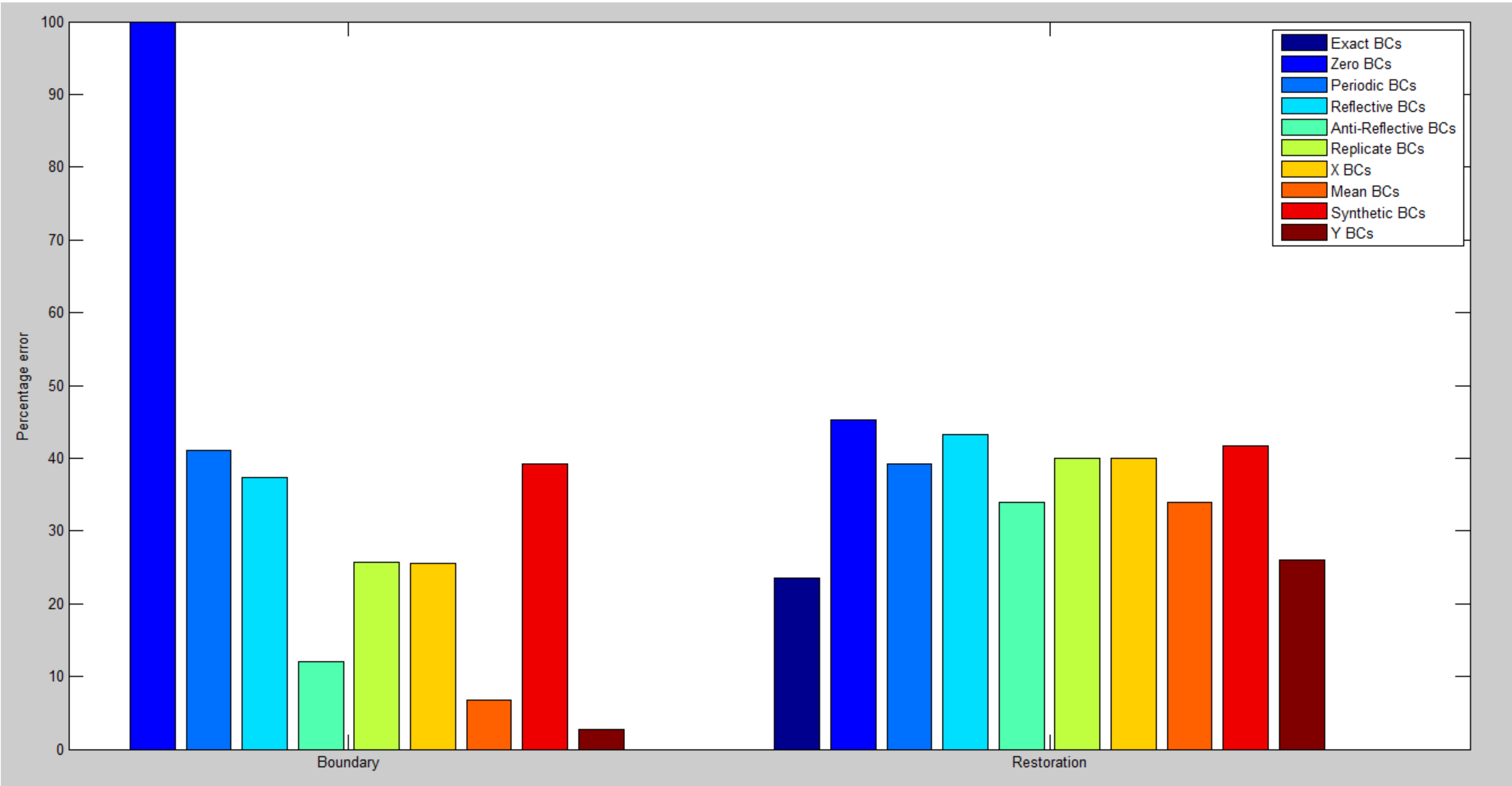
where b_{ext} is the double-size extension of b, obtained following the BCs imposed, and C is the BCCB matrix associated with the double-size extension of the original PSF of the problem, obtained by a pad array of zeros. Clearly in this case the restored image will be the central part of x_{ext} corresponding to b.

Test 1: Gaussian blur





Test 1: Gaussian blur



Test 2: motion blur

Test2: motion blur