IMAGE DEBLURRING - COMPUTATION OF CONFIDENCE INTERVALS

VICTORIA TAROUDAKI
Prof. Dianne P. O'Leary

UNIVERSITY OF MARYLAND

MAY 3, 2011

Outline

Blurred Images are everywhere

(a) Actual scene

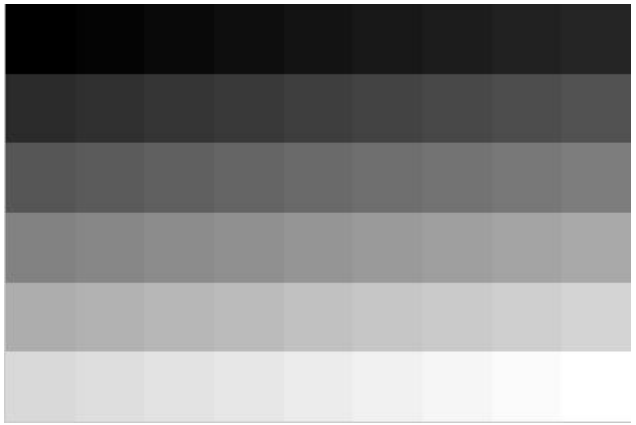
(b) Blurred Image

(c) Ideal recovered image

The image

- An image is an array of pixels
- For a grayscale image, these values are in the interval [0,255]

$$0 = \textit{black}$$
$$255 = \textit{white}$$



The problem

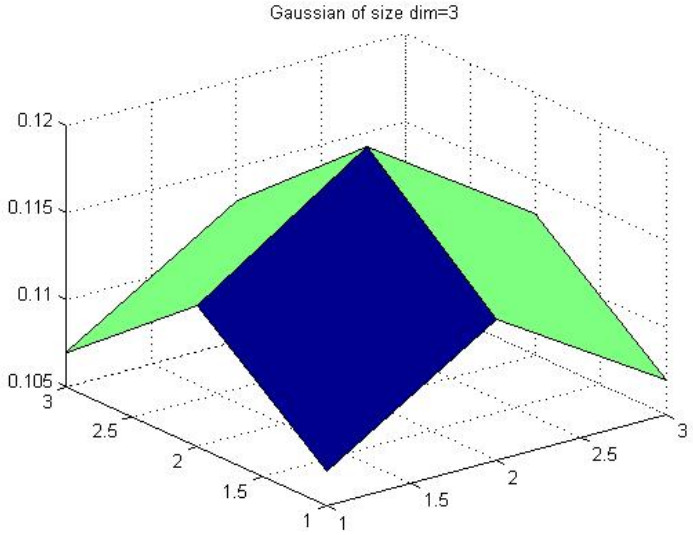
- Notation:

Symbol	Size	Explanation
K	$m \times n$	Matrix defined through the Point Spread function (PSF) in the case of a linear problem
X		Original Clear Image
x	$n \times 1$	Vector containing the values corresponding to the pixels of the image X
B		The blurred image we measure
b	$m \times 1$	Vector which contains the values of the pixels of the blurred image B
e	$m \times 1$	Noise Vector

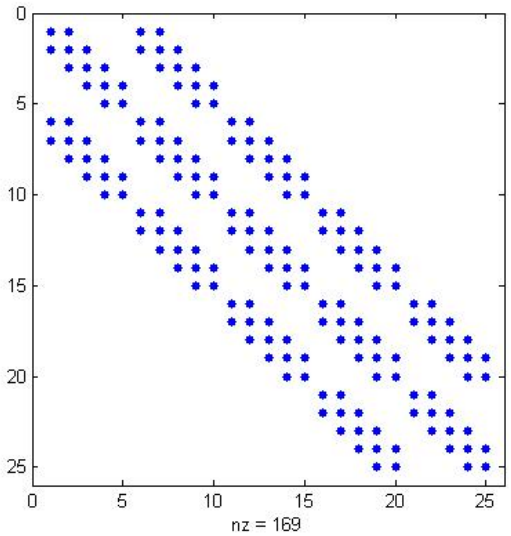
- $b = Kx + e$

PSF-Blurring matrix

- The Point Spread Function (PSF) used is the Gaussian.
- The blurring matrix is block diagonal with so many diagonal blocks as the number of columns of the PSF matrix.
- Example for a 3×3 PSF with a 5×5 image:



(d) Gaussian PSF of size 3×3



(e) Blurring matrix of an image 5×5 using a PSF 3×3

Goal

Computation of Confidence Intervals for Images blurred by

$$\mathbf{b}=\mathbf{K}\mathbf{x}+\mathbf{e}$$

Confidence Intervals

Definition

One-at-a-time confidence intervals bound each φ_k individually with probability α (α%confidence).

$$Pr\{l_k \leq \varphi_k \leq u_k\} = \alpha, k = 1, 2, \ldots, p$$

Definition

Simultaneous confidence intervals which bound all the φ_k simultaneously with a probability greater than or equal to α.

$$Pr\{l_k \leq \varphi_k \leq u_k, k = 1, 2, \ldots, p\} \geq \alpha$$

Assumptions- Notation

Assumptions

- Noise (e) normally distributed with mean zero and standard deviation S. ($e \sim \mathcal{N}(0, S^2)$)
- S nonsingular and symmetric

Notation

- lb_i The lower bound of the confidence interval corresponding to the pixel i
- ub_i The upper bound of the confidence interval corresponding to the pixel i

Serial Algorithm - The image as a Whole

Algorithm

for *each pixel i of the image*

$$\textit{solve } min\{w_k^T x : \|Kx - b\|_S \leq \mu, 0 \leq x \leq 255\} \textit{ for the } lb_i$$

$$\textit{solve } max\{w_k^T x : \|Kx - b\|_S \leq \mu, 0 \leq x \leq 255\} \textit{ for the } ub_i$$

end

Parallel Algorithm - The image as a Whole

Algorithm

parfor *each pixel i of the image*

$$\textit{solve } min\{w_k^T x : \|Kx - b\|_S \leq \mu, 0 \leq x \leq 255\} \textit{ for the } lb_i$$

$$\textit{solve } max\{w_k^T x : \|Kx - b\|_S \leq \mu, 0 \leq x \leq 255\} \textit{ for the } ub_i$$

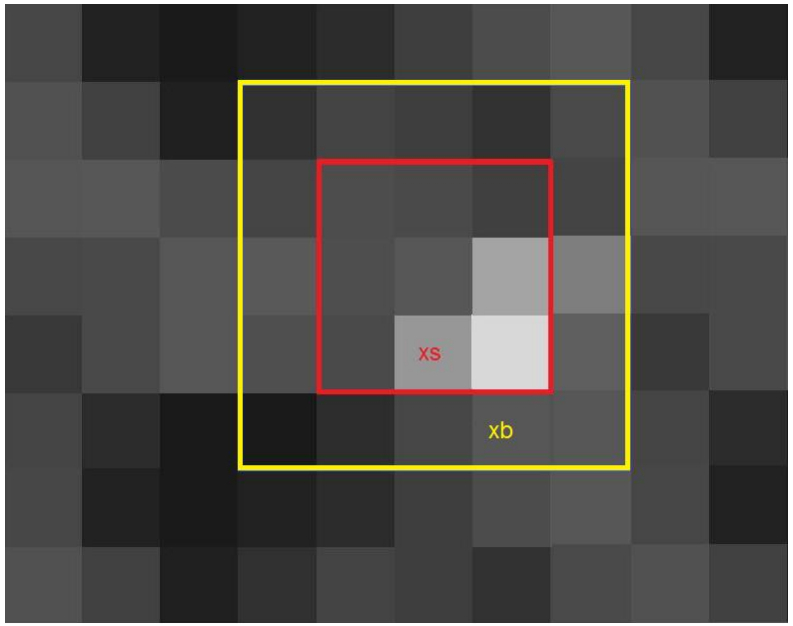
end

Main problem of the full image approach

- For each one of the pixels in the image we use iterative methods for the minimization (or maximization) of the norms.
- These methods use the blurring matrix K which for an $n \times n$ image is of size $n^2 \times n^2$ (e.g. for a 256×256 image, K is of size 65536×65536).

To deal with this problem, we use another approach, that of the sub-images and the sub-matrices.

Construction of Sub-Images



x_s the sub-image of desirable size
x_b the boundary of appropriate size

Notation for the sub-problems

- Notation:

Symbol	Size	Explanation
K_s	$rc \times rc$	Part of the blurring matrix K that corresponds to the subimage
X_s	$r \times c$	Original Clear Sub-Image
x_s	$rc \times 1$	Vector containing the values corresponding to the pixels of the sub-image X_s
x_t		The vector of values of the rest of the pixels in the image
K_t		Part of the blurring matrix K that corresponds to x_t
b_s	$rc \times 1$	Vector which contains the values of the pixels of the blurred image B_s

- $b_s - K_t x_t = K_s x_s \leftrightarrow b_{st} = K_s x_s$

Serial Algorithm - Use of Sub-Images

Algorithm

for *each sub-image s*

  for *each pixel i of the sub-image*

    solve $min\{w_k^T x_s : \|K_s x_s - b_{st}\|_S \leq \mu, 0 \leq x_s \leq 255\}$ *for the* lb_i

    solve $max\{w_k^T x_s : \|K_s x_s - b_{st}\|_S \leq \mu, 0 \leq x_s \leq 255\}$ *for the* ub_i

  end

end

Parallel Algorithm - Use of Sub-Images

Algorithm

for *each sub-image s*

parfor *each pixel i of the sub-image*

solve $min\{w_k^T x_s : \|K_s x_s - b_{st}\|_S \leq \mu, 0 \leq x_s \leq 255\}$ *for the* lb_i

solve $max\{w_k^T x_s : \|K_s x_s - b_{st}\|_S \leq \mu, 0 \leq x_s \leq 255\}$ *for the* ub_i

end

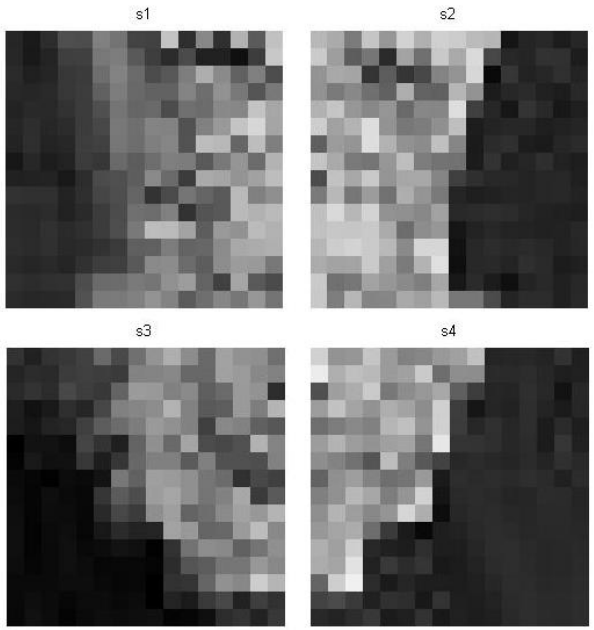
end

Database

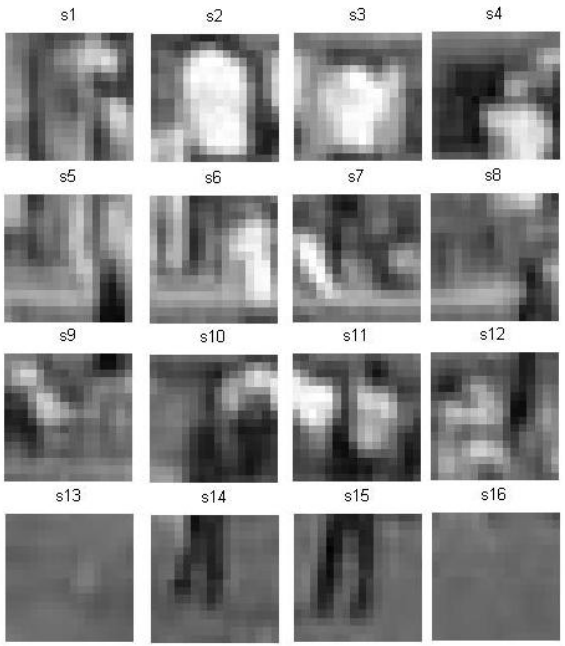
Gray-scale images of various sizes.

Examples of test images:

Example of sub-images



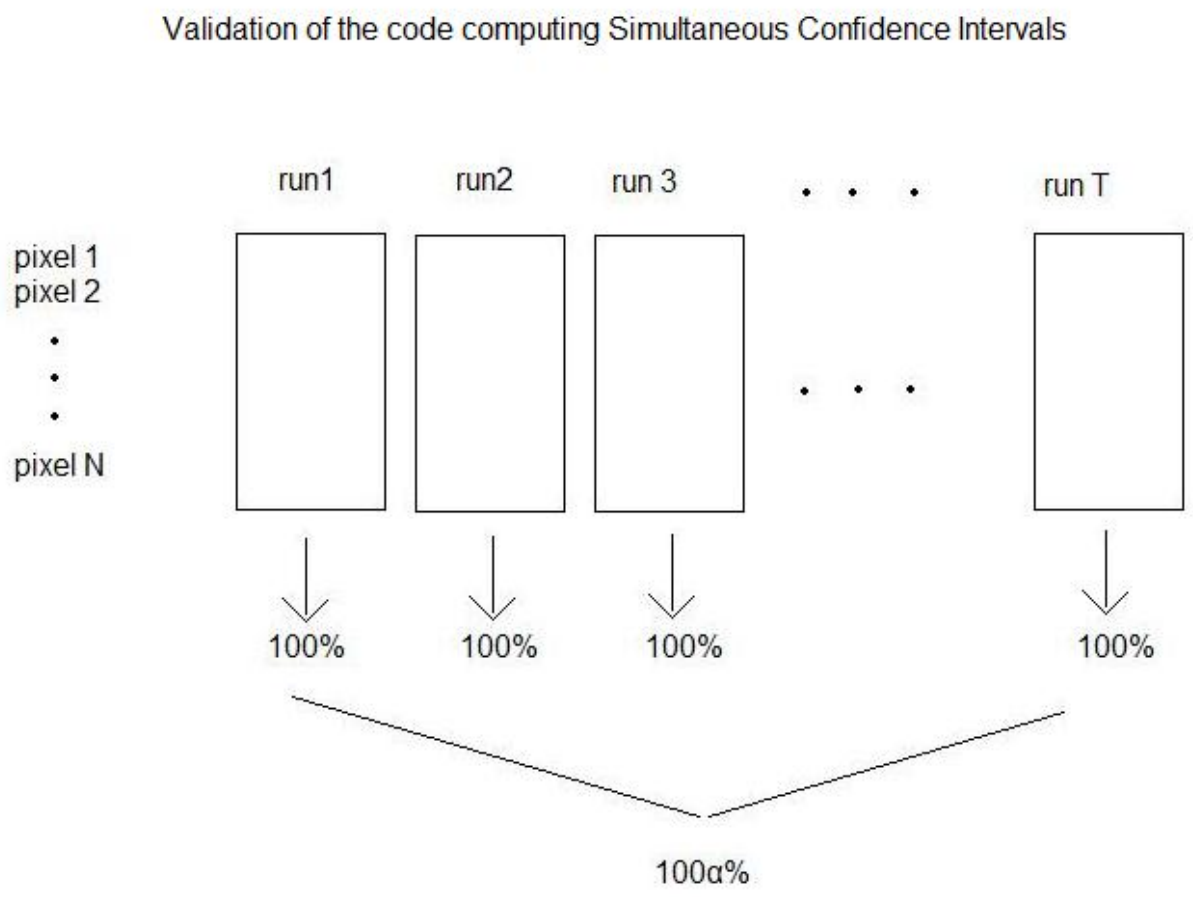
(f) 4 16×16 sub-images



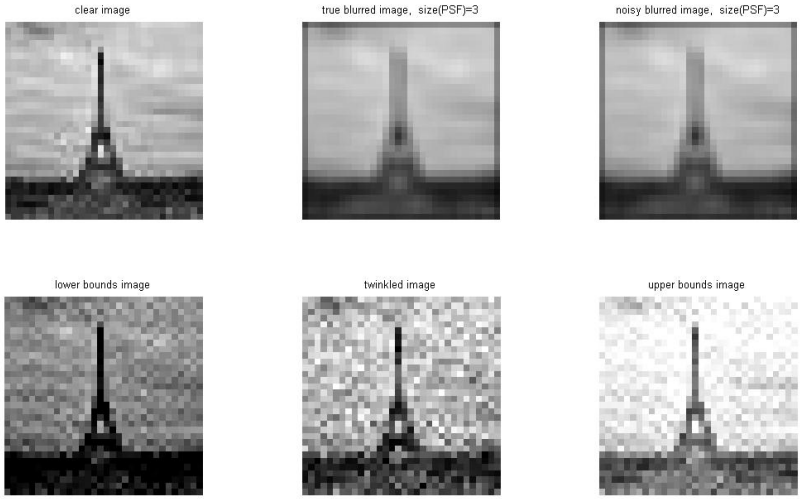
(g) 8 16×16 sub-images

Validation

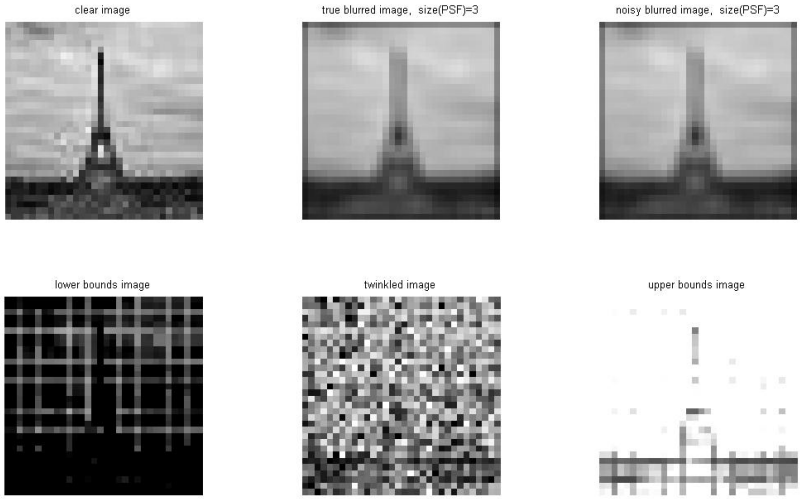
For $\alpha\%$ confidence interval with $(0 < \alpha < 100)$ the validation is done by multiple runs of the code using the same data and values of parameters.



"Eiffel Tower 32", subimages 4×4



"Eiffel Tower 32", subimages 8×8



Boundary

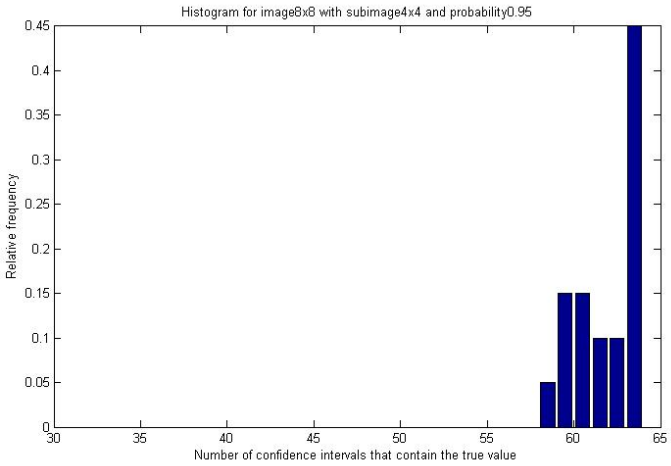
- Formula to compute the number of pixels in the boundary (BP) of a subimage given its size ($n \times n$) and the size of the PSF ($p \times p$) is the following

$$BP = 4\left(n + \frac{p-1}{2}\right)\frac{p-1}{2} = 2(p-1)\left(n + \frac{p-1}{2}\right)$$

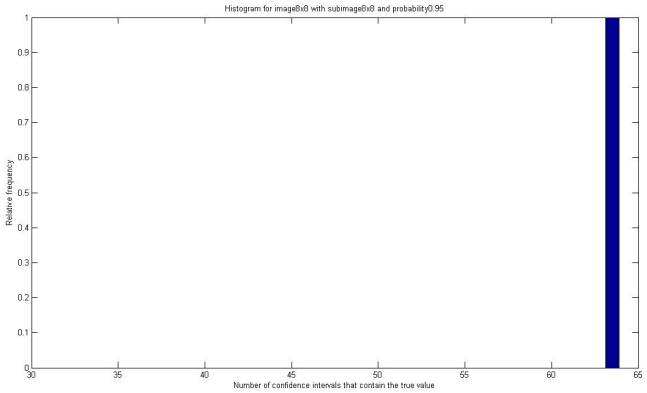
- Table with the number of the pixels on the boundary for several subimage and PSF sizes.

Subimage Size \ PSF size		3	5	7	9	11	13
16	4×4	20	48	84	128	180	240
64	8×8	36	80	132	192	260	336
256	16×16	68	144	228	320	420	528
1024	32×32	132	272	420	576	740	912
4096	64×64	260	528	804	1088	1380	1680

Validation



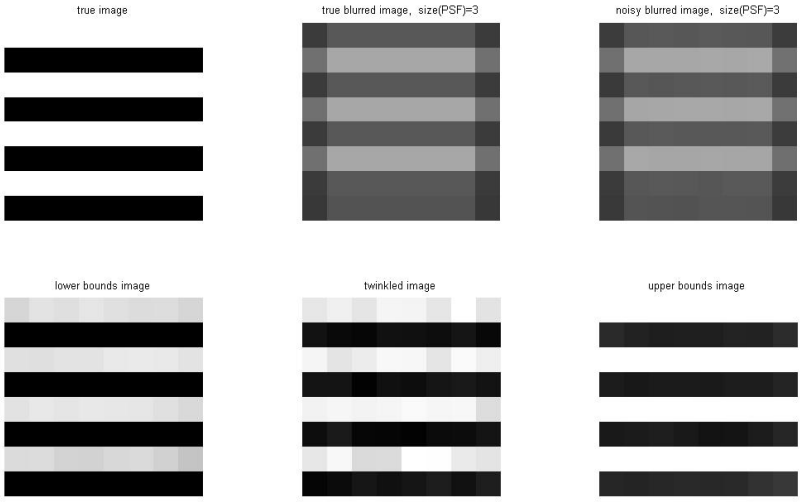
(h) Histogram with subimages 4×4



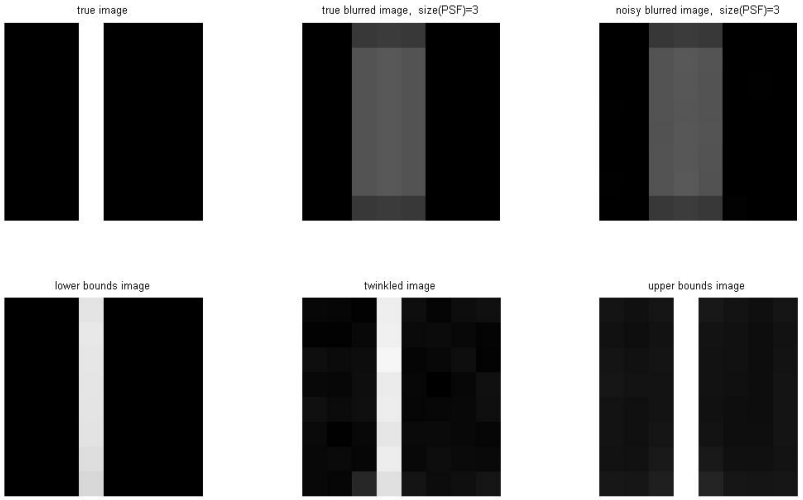
(i) Histogram with subimages 8×8

Figure: Validation Histograms.

"stripes 8" image, fp

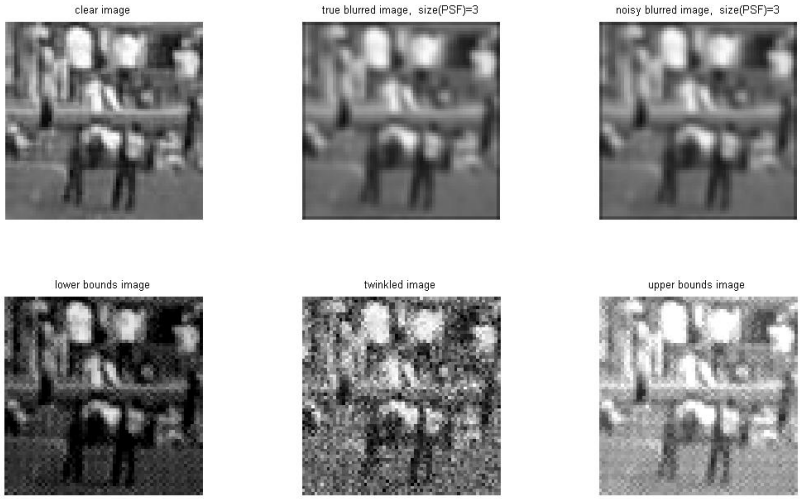


"vertical line 8" image, fs

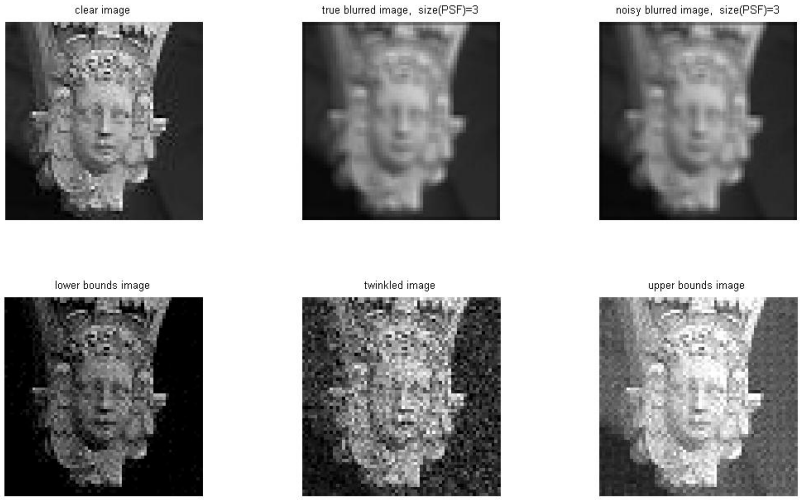


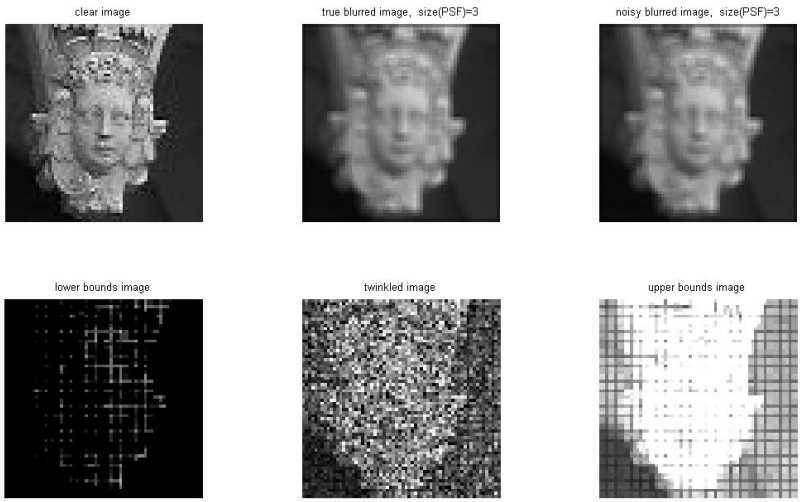
"Capitol 32" image, 4s

"Capitol 64" image, 4p

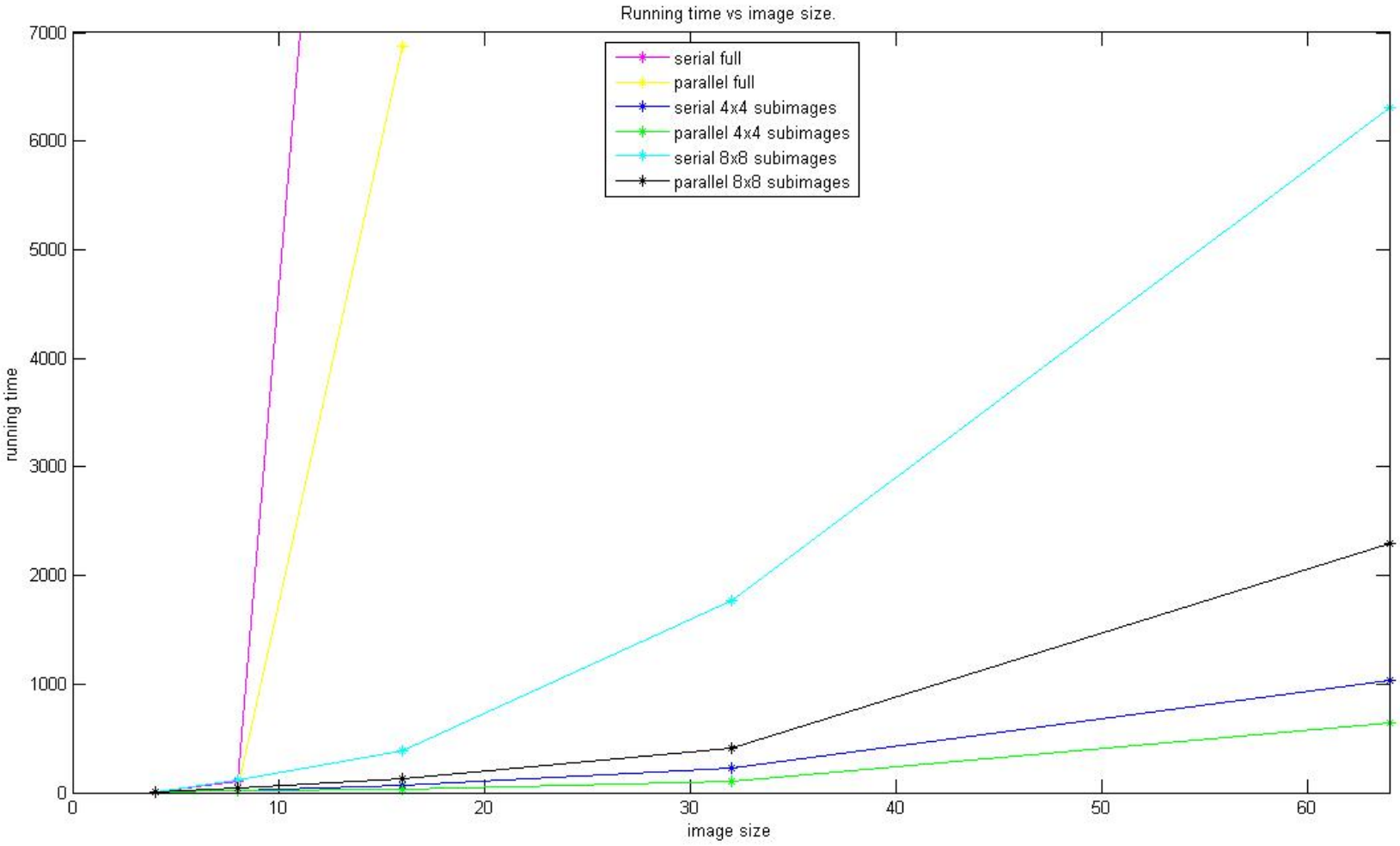


"La Rochelle 64" image, 4s and 8p

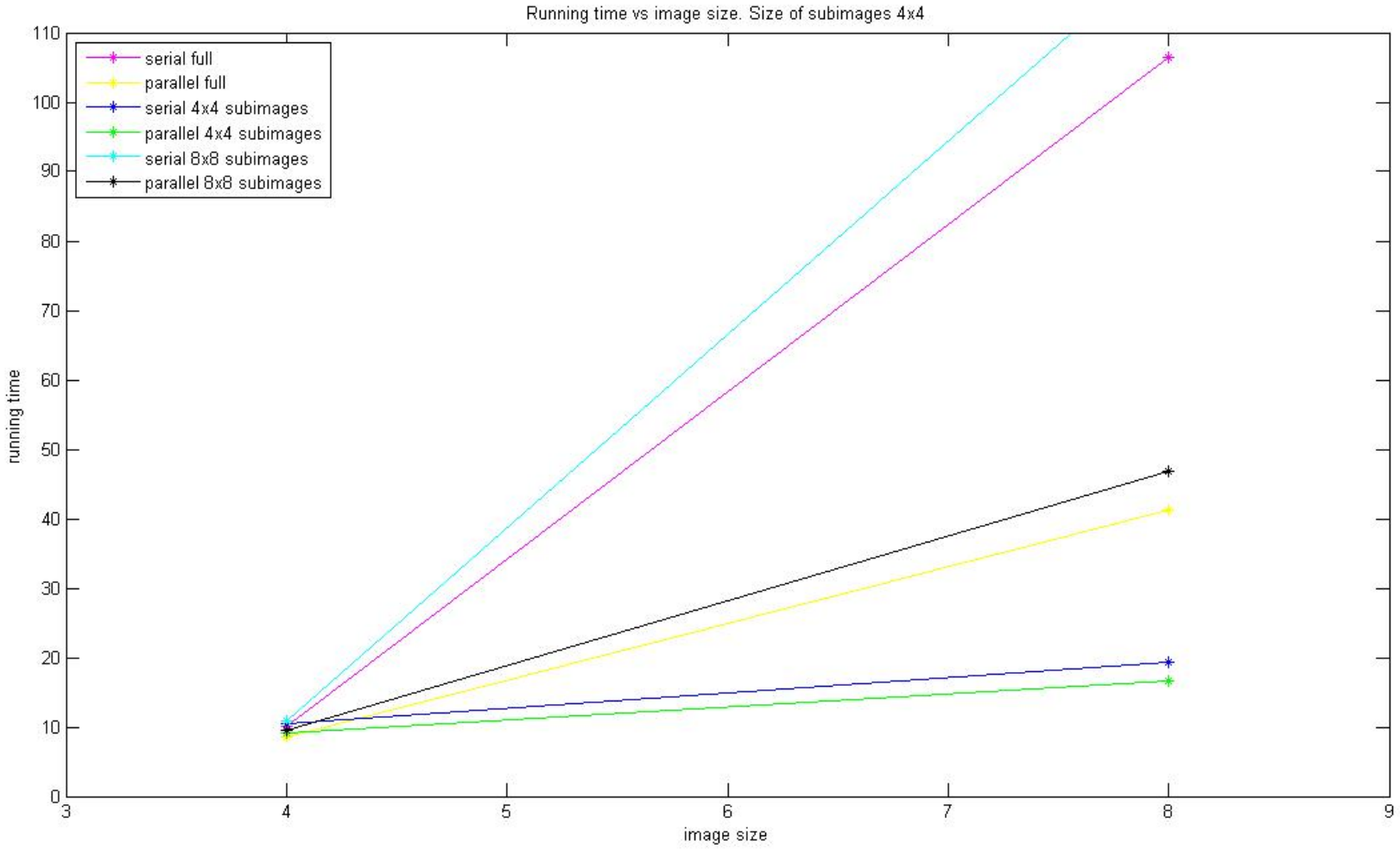




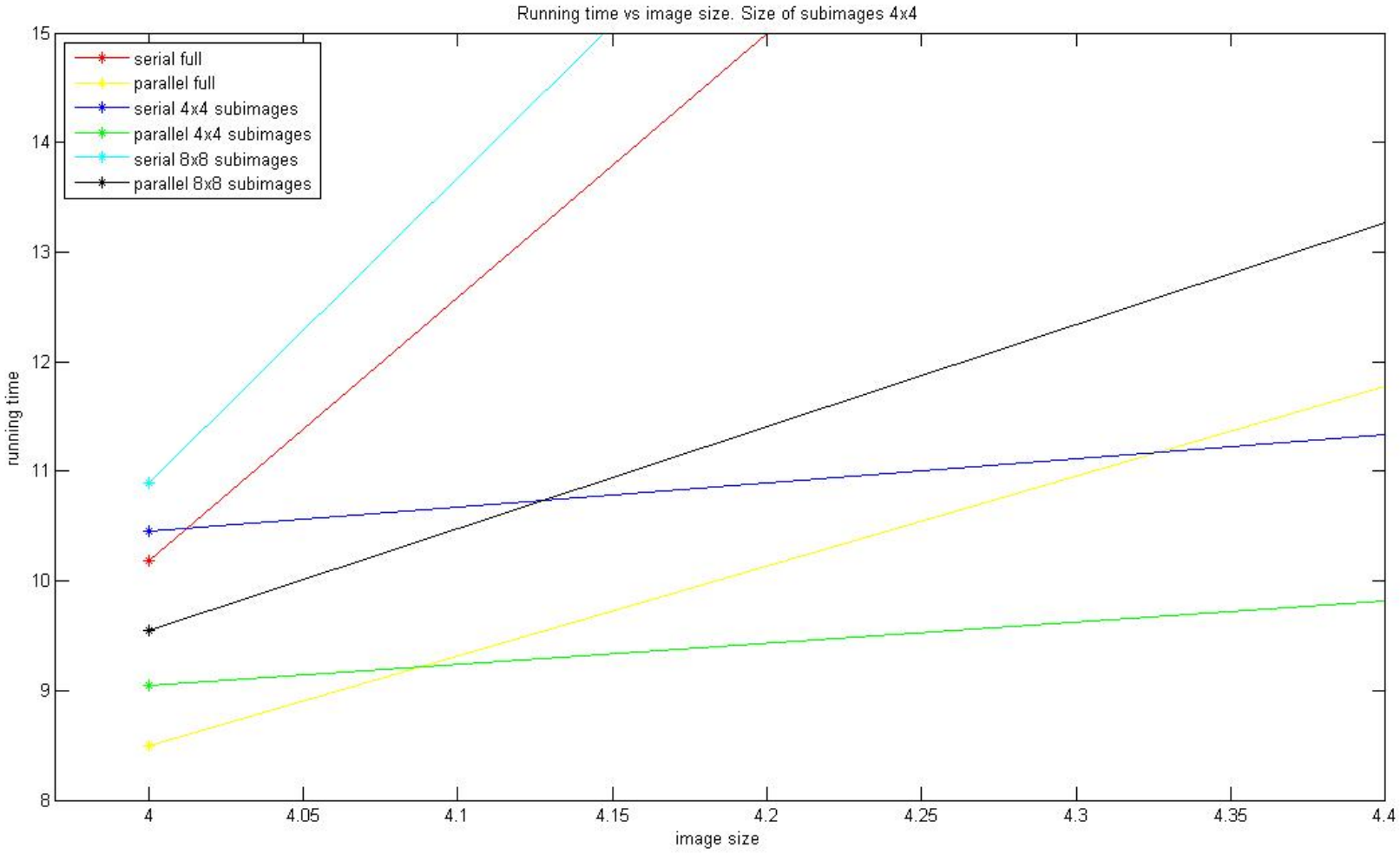
Size of image vs running time



Size of image vs running time (8×8)



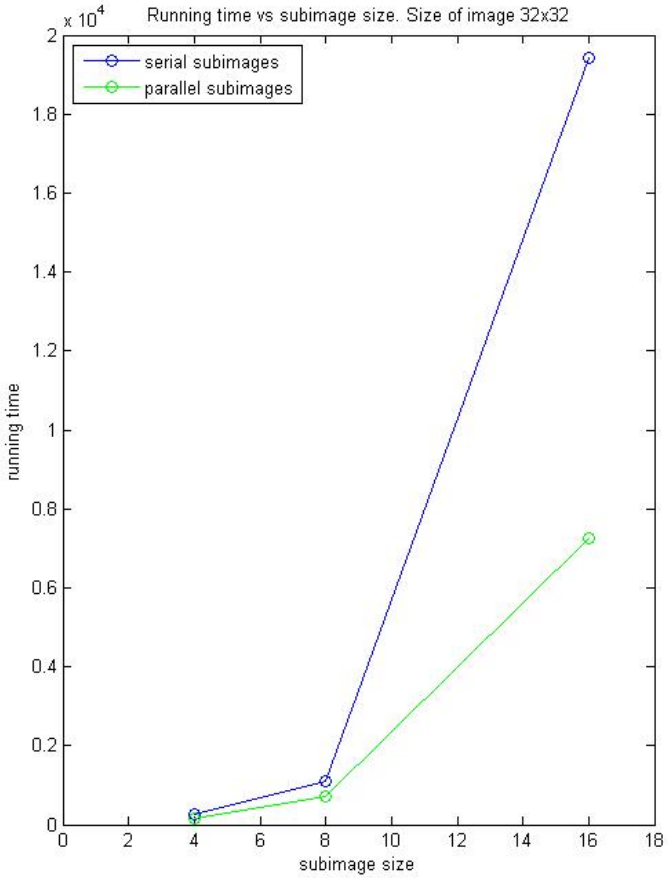
Size of image vs running time (4×4)

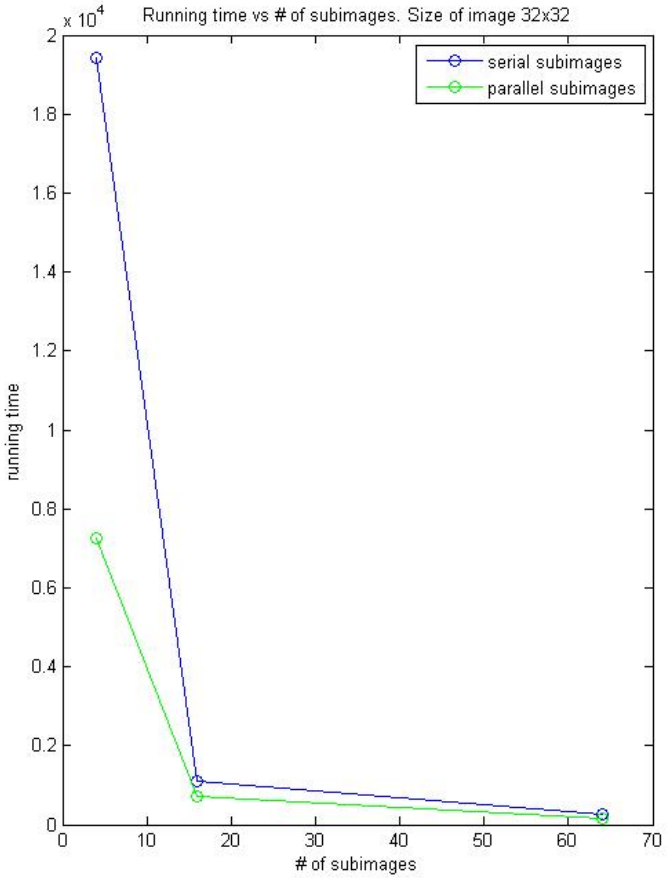


Conclusions

- The parallel codes work better than the corresponding serial ones.
- The sub-images code is quicker than the full image except for the case when the codes deal with the image on the same way, i.e., when the sub-image is of the same size of the original image.

Size and number of sub-images vs running time





Conclusions

- When we double the size of the sub-images, the number of them is divided by 4.
- With increasing size of the sub-images, the running time increases.
- Size of sub-images more dominant than their number.

Schedule-Milestones

Phase I: Serial Code for the computation of the confidence intervals

Finished

Phase II:Parallel Code for computing the confidence intervals using sub-images and sub-matrices

Finished

Deliverables

- Code
  - Full-image serial code
  - Full-image parallel code
  - Sub-images serial code
  - Sub-images parallel code
- Database images
- Validation module
- Report

Summary

- Code that computes the confidence intervals of a whole image and of sub-images
- Parallelization of the codes to improve speed
- Experiment with different types and sizes of images, different sizes of PSF functions
- Comparison of the results with respect to time

Results showed that the parallelized method of computing the confidence intervals using sub-images works the best

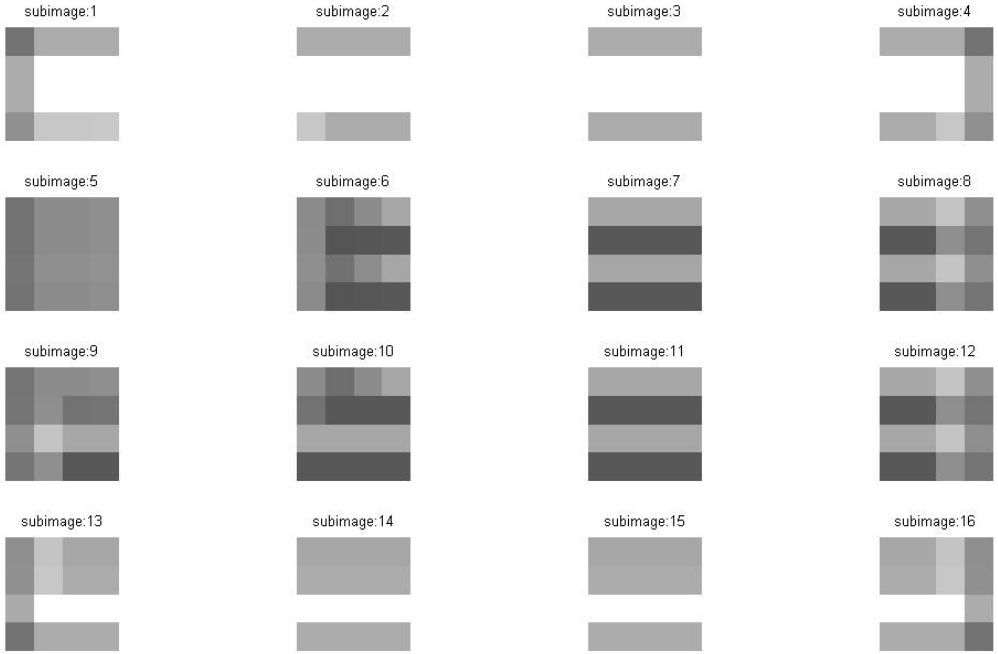
Bibliography

- Tony F. Chan and Jianhong (Jackie) Shen, *"Image Processing and Analysis"*, SIAM, Philadelphia, 2005
- Martin Hanke, James Nagy and Robert Plemmons, *"Preconditioned Iterative Regularization For Ill-Posed Problems"*, IMA Preprint Series n 1024, 1992
- Per Christian Hansen, James G. Nagy and Dianne P. O'Leary, *"Deblurring Images Matrices, Spectra, and Filtering"*, SIAM, Philadelphia, 2006
- Richard A. Johnson, Gouri K. Bhattacharyya, *"Statistics: Principles and Methods"*, John Wiley & Sons, Inc., 2006
- Charles L. Lawson and Richard J. Hanson, *"Solving Least Squares problems"*, SIAM, Philadelphia, 1995
- Jodi L. Mead, Rosemary A Renaut, *"Least squares problems with inequality constraints as quadratic constraints"*,Linear Algebra and its Applications, 432, 2010, p. 1936ffl1949

Bibliography

- James G. Nagy and Dianne P. O'Leary, *"Restoring Images Degraded By Spatially- Variant Blur"*, SIAM J. Sci. Comput., Vol 19, No 4, 1998, p. 1063-1082
- James G. Nagy and Dianne P. O'Leary, *"Image Restoration through Subimages and Confidence Images"*, Electronic Transactions on Numerical Analysis, 13, 2002, p. $22 - 37$
- Dianne P. O'Leary and Bert W. Rust, *"Confidence Intervals for inequality constrained least squares problems, with applications to ill-posed problems"*, SIAM Journal on Scientific and Statistical Computing, 7, 1986, p. $473 - 489$
- Bert W. Rust and Dianne P. O'Leary, *"Confidence intervals for discrete approximations to ill-posed problems"*, The Journal of Computational and Graphical Statistics, 3, 1994, p. $67 - 96$
- L. Tenorio, A.Fleck and K. Moses, *"Confidence intervals for linear discrete inverse problems with non negativity constraint"*, Inverse Problems, 23, 2007, p. 669-681

Thank you


subimage:1
subimage:2
subimage:3
subimage:4
subimage:5
subimage:6
subimage:7
subimage:8
subimage:9
subimage:10
subimage:11
subimage:12
subimage:13
subimage:14
subimage:15
subimage:16