Computer Vision and EM

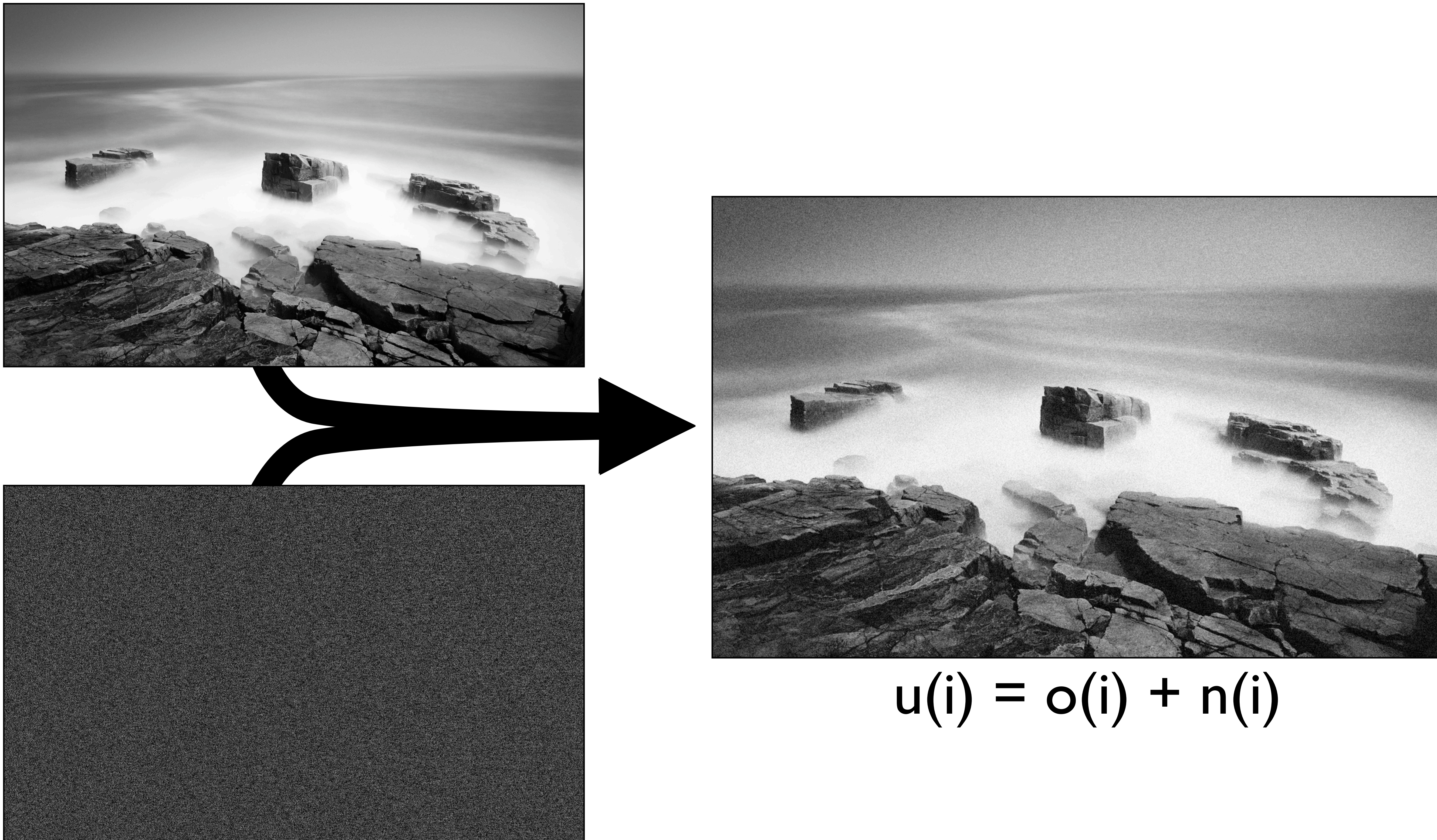
Time Complexity

	n	nlgn	n^2	n^3	1.5^n	2^n	n!
n = 10	<1 sec	<1 sec	<1 sec	<1 sec	<1 sec	<1 sec	4 sec
n = 30	<1 sec	<1 sec	<1 sec	<1 sec	<1 sec	18 min	10^{25} years
n = 50	<1 sec	<1 sec	<1 sec	<1 sec	11 min	36 years	very long
n = 100	<1 sec	<1 sec	<1 sec	1 sec	12,892 years	10^{17} years	very long
n = 1,000	<1 sec	<1 sec	1 sec	18 min	very long	very long	very long
n = 10,000	<1 sec	<1 sec	2 min	12 days	very long	very long	very long
n = 100,000	<1 sec	2 sec	3 hours	32 years	very long	very long	very long
n = 1,000,000	1 sec	20 sec	12 days	31,710 years	very long	very long	very long

Adapted from Algorithm Design, Kleinberg and Tardos

very long: greater than 10^{25} years
age of the universe: 1.37×10^{10} years
p = np?: priceless

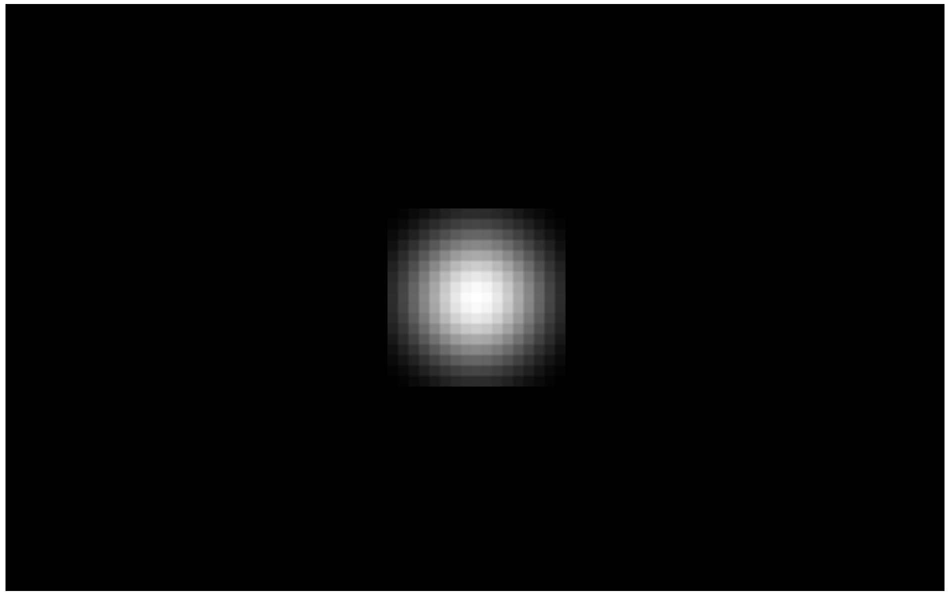
Denoising



$u(i) = o(i) + n(i)$

Gaussian Blur

$*$



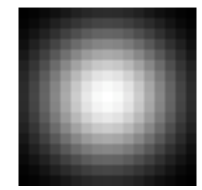
$$G(x)=Ne^{\frac{x^2}{2\sigma^2}}$$

$$N=\frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}}$$

```
x = distance from zero
σ = sigma
N = normalization factor
n = number of dimensions
```

Gaussian Blur

$*$

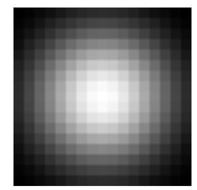


$$G(x)=Ne^{\frac{x^2}{2\sigma^2}}$$

$$N=\frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}}$$

```
x = distance from zero
σ = sigma
N = normalization factor
n = number of dimensions
```

Gaussian Blur

 * 

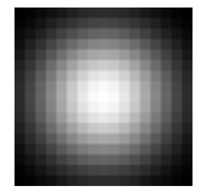
$$G(x)=Ne^{\frac{x^2}{2\sigma^2}}$$

$$N=\frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}}$$

```
x = distance from zero
σ = sigma
N = normalization factor
n = number of dimensions
```

Gaussian Blur

$*$



=

$$G(x)=Ne^{\frac{x^2}{2\sigma^2}}$$

$$N=\frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}}$$

```
x = distance from zero
σ = sigma
N = normalization factor
n = number of dimensions
```

Gaussian Blur

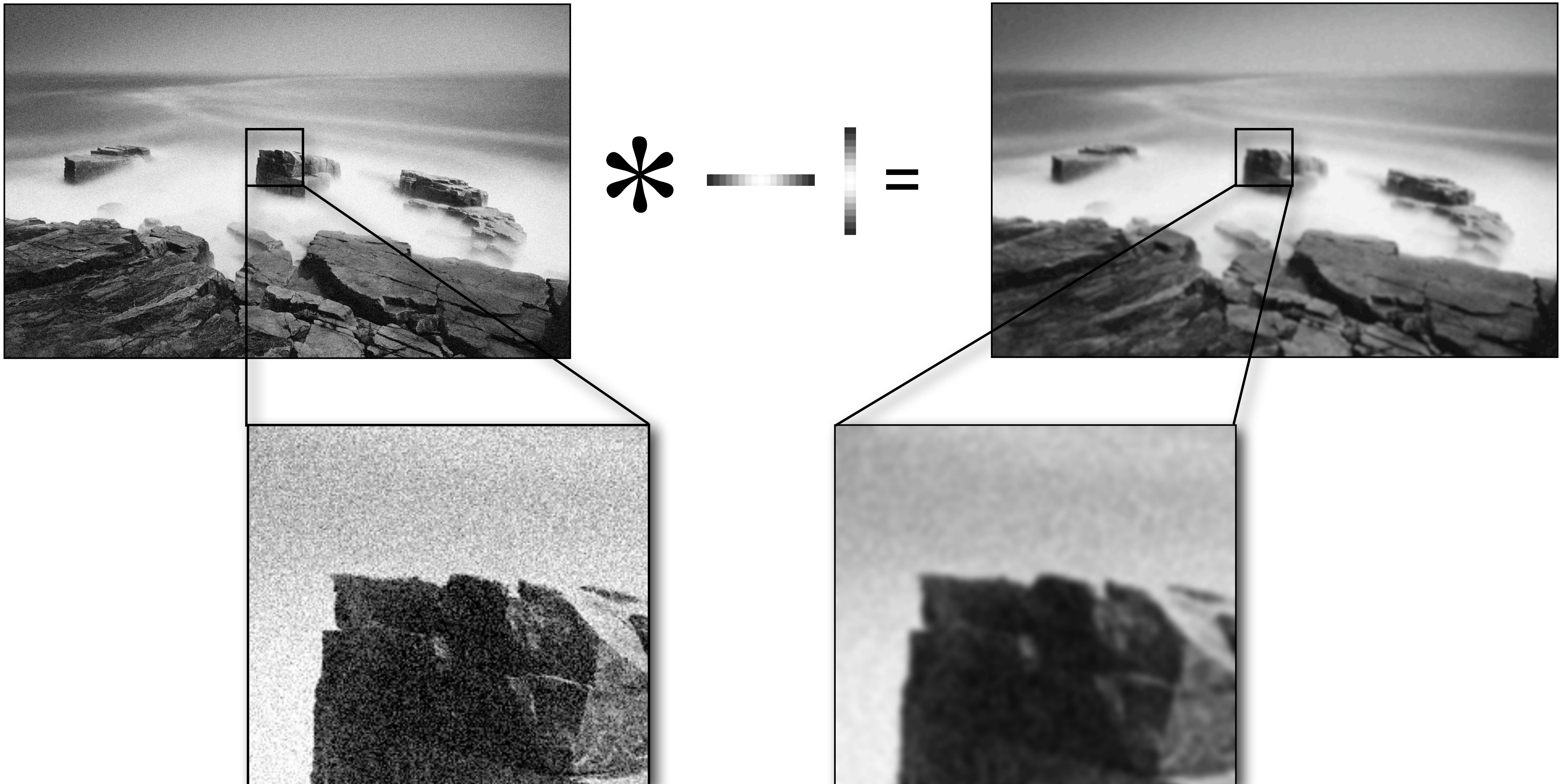
 $*$  $=$ 

$$G(x) = Ne^{\frac{x^2}{2\sigma^2}}$$

$$N = \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}}$$

```
x = distance from zero
σ = sigma
N = normalization factor
n = number of dimensions
```

Gaussian Blur



Median Filter

Median Filter

104	107	99
4	3	3
4	5	4

Median Filter

104	107	99
4	3	3
4	5	4

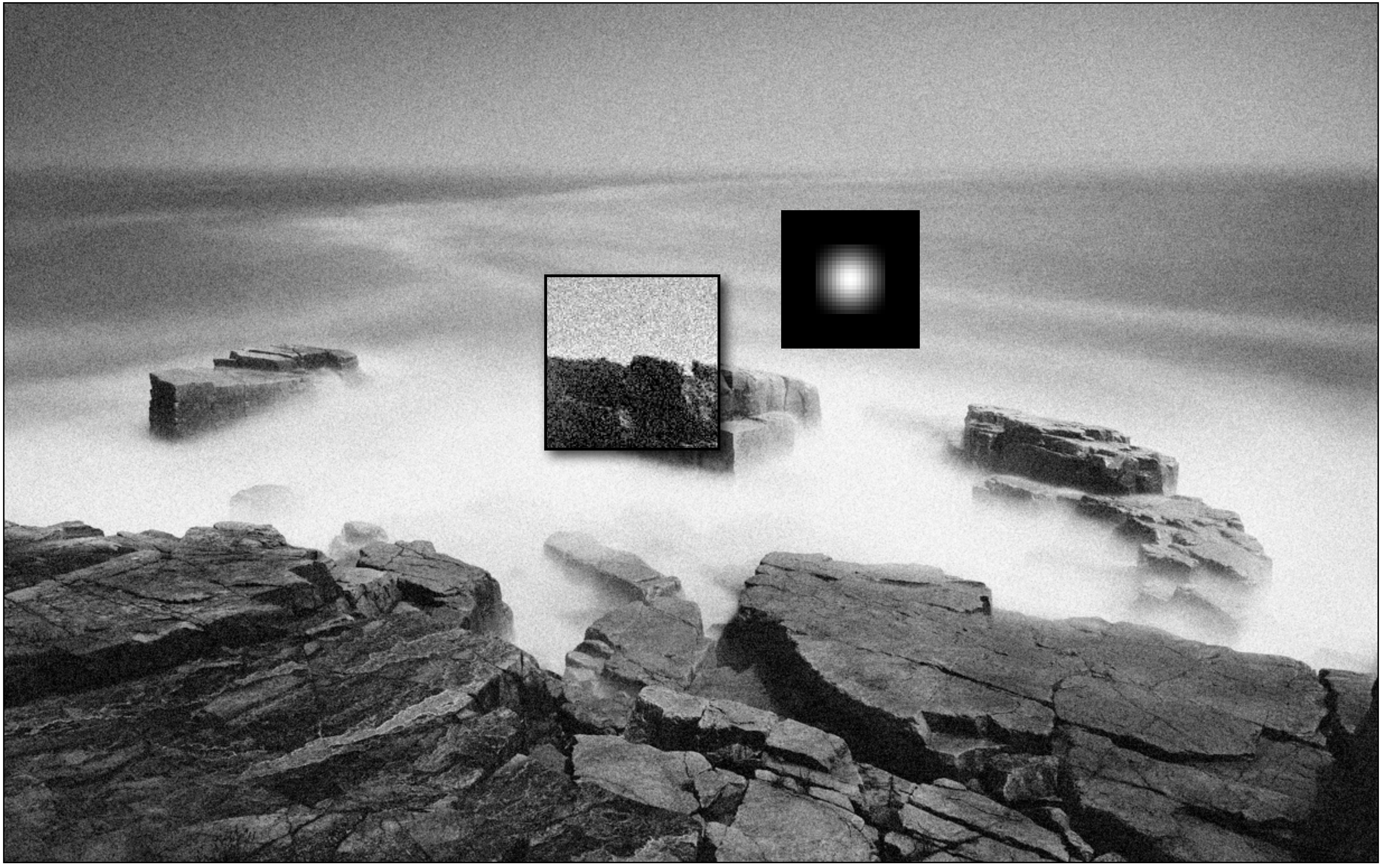
3	3	4
4	4	5
104	107	99

Median Filter

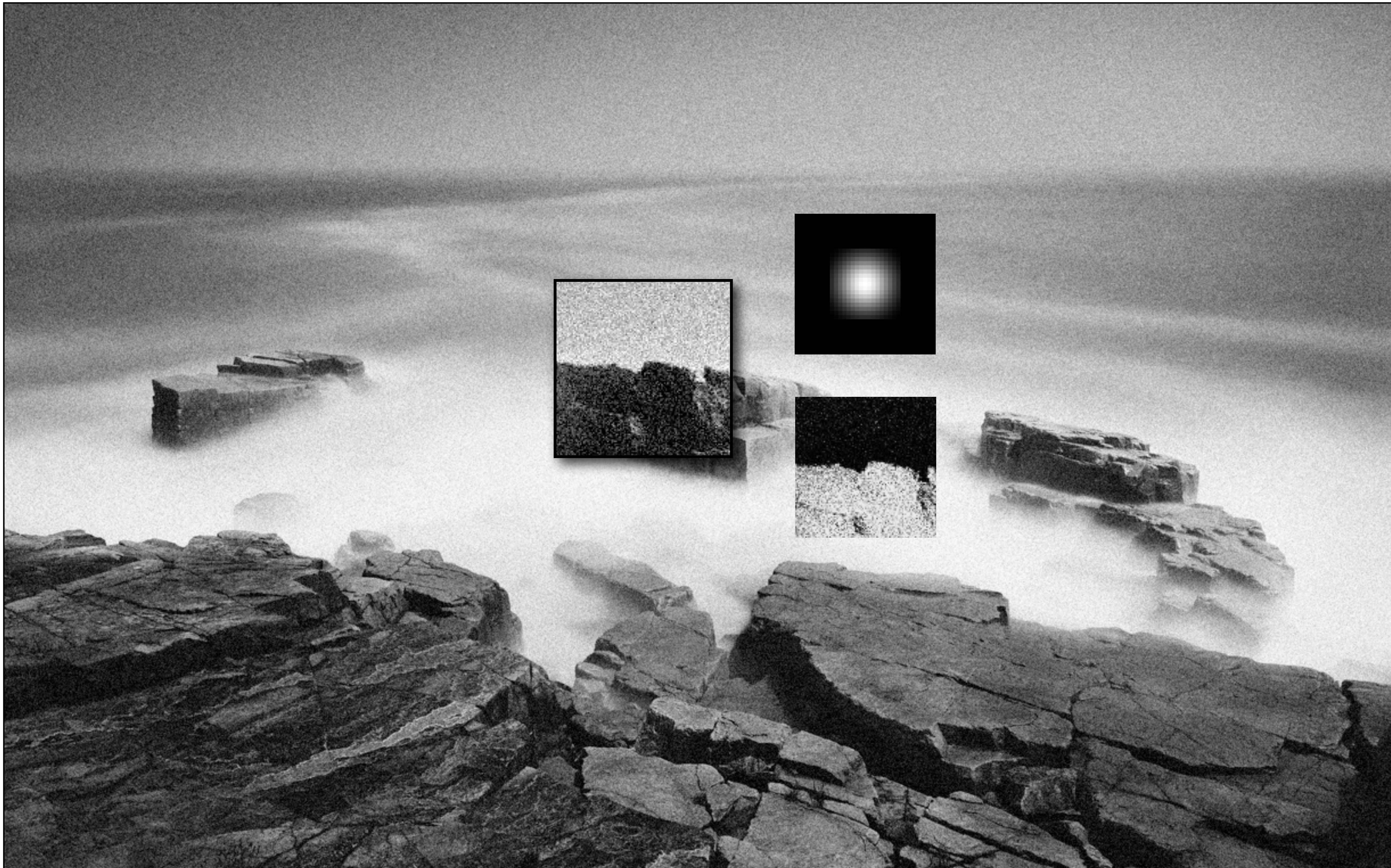
Bilateral Filter

Bilateral Filter

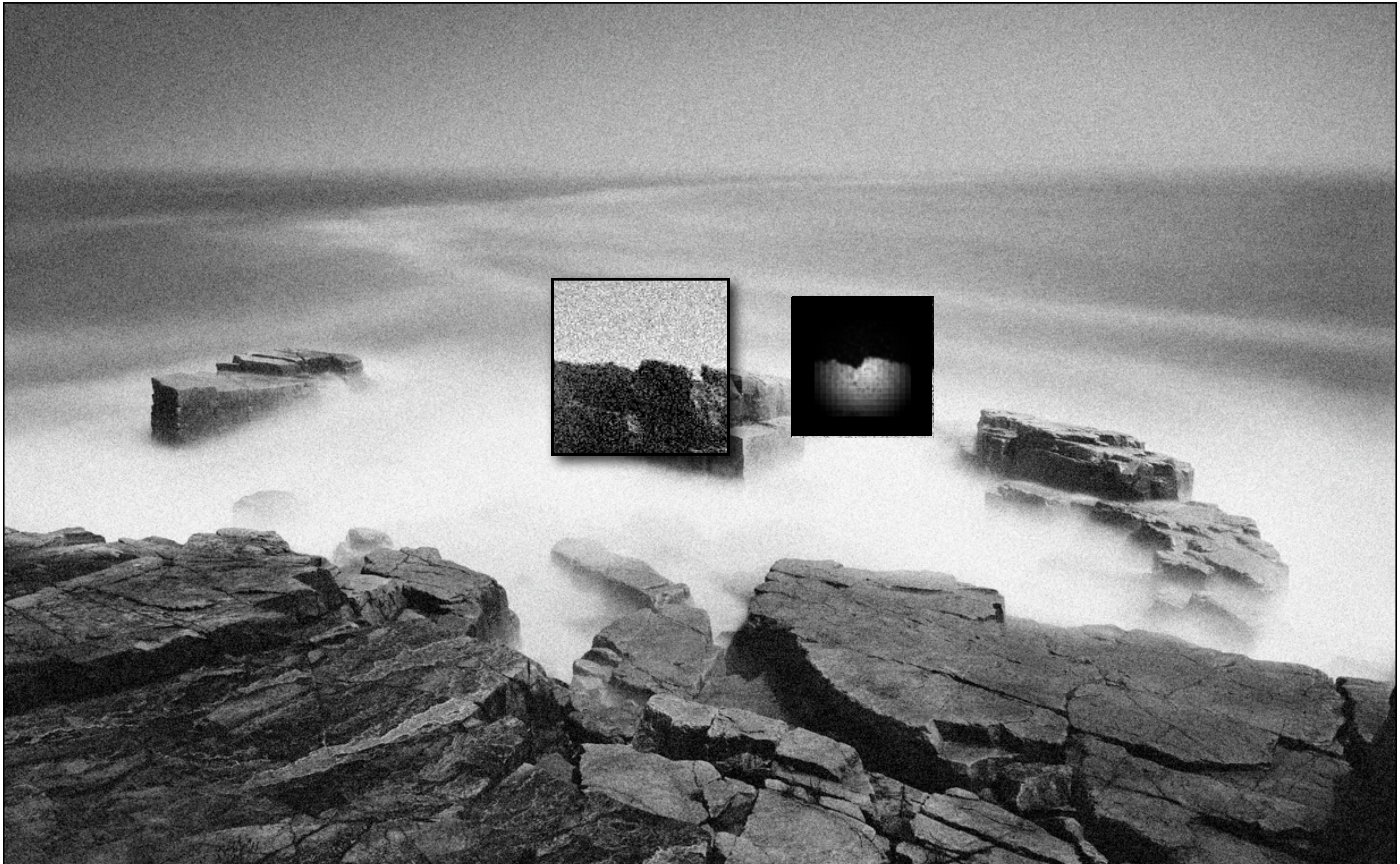
Bilateral Filter



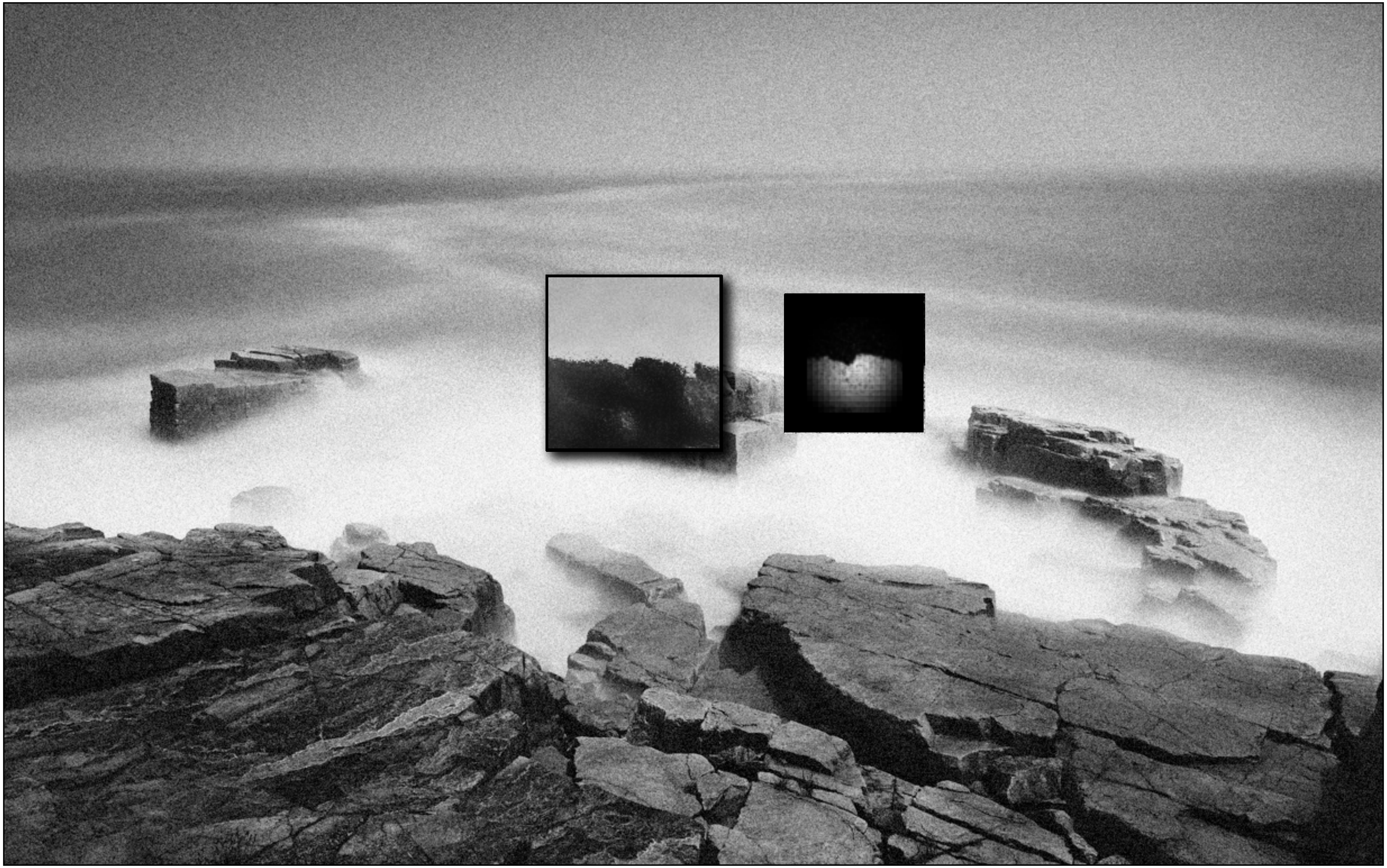
Bilateral Filter



Bilateral Filter



Bilateral Filter



Bilateral Filter

Bilateral Filter

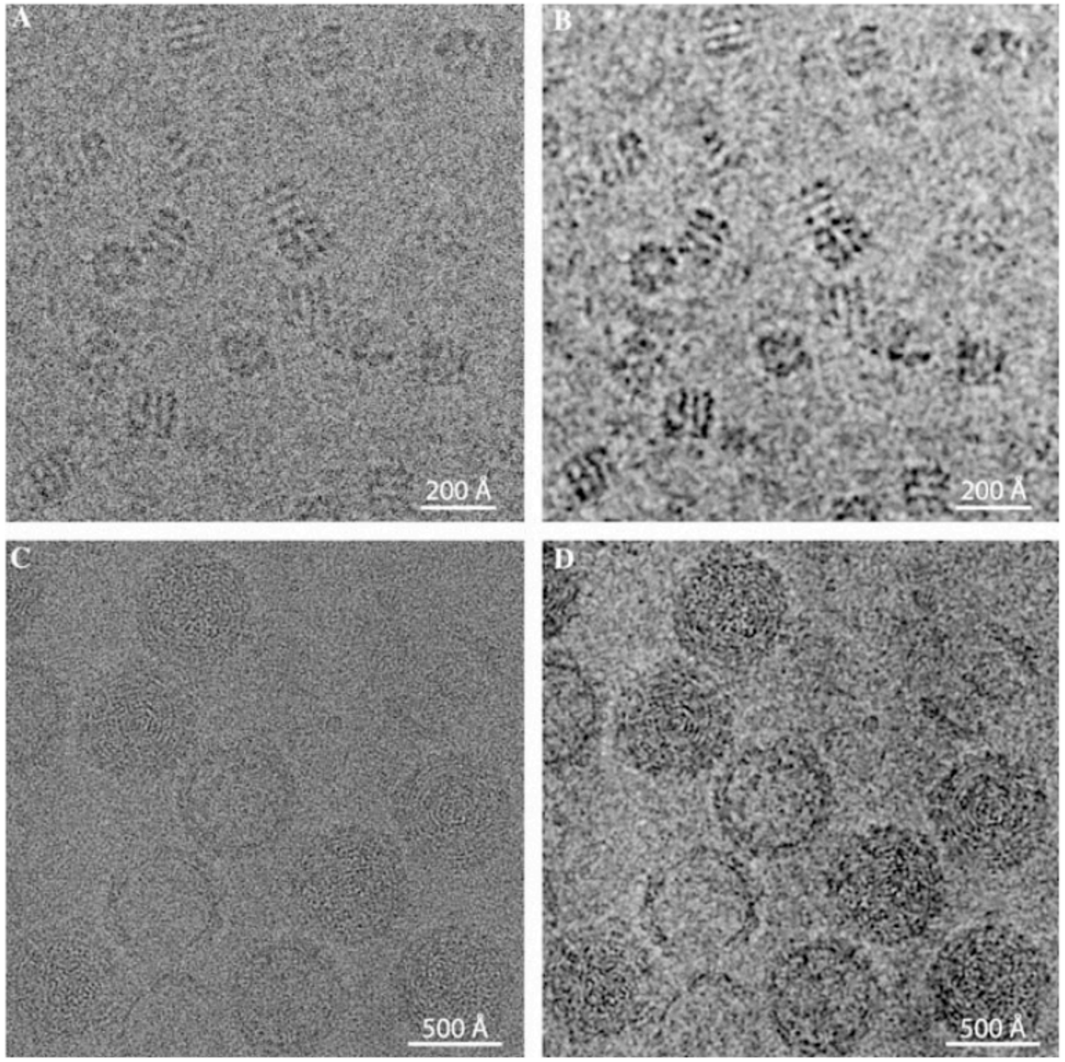
Applications of a bilateral denoising filter in biological electron microscopy

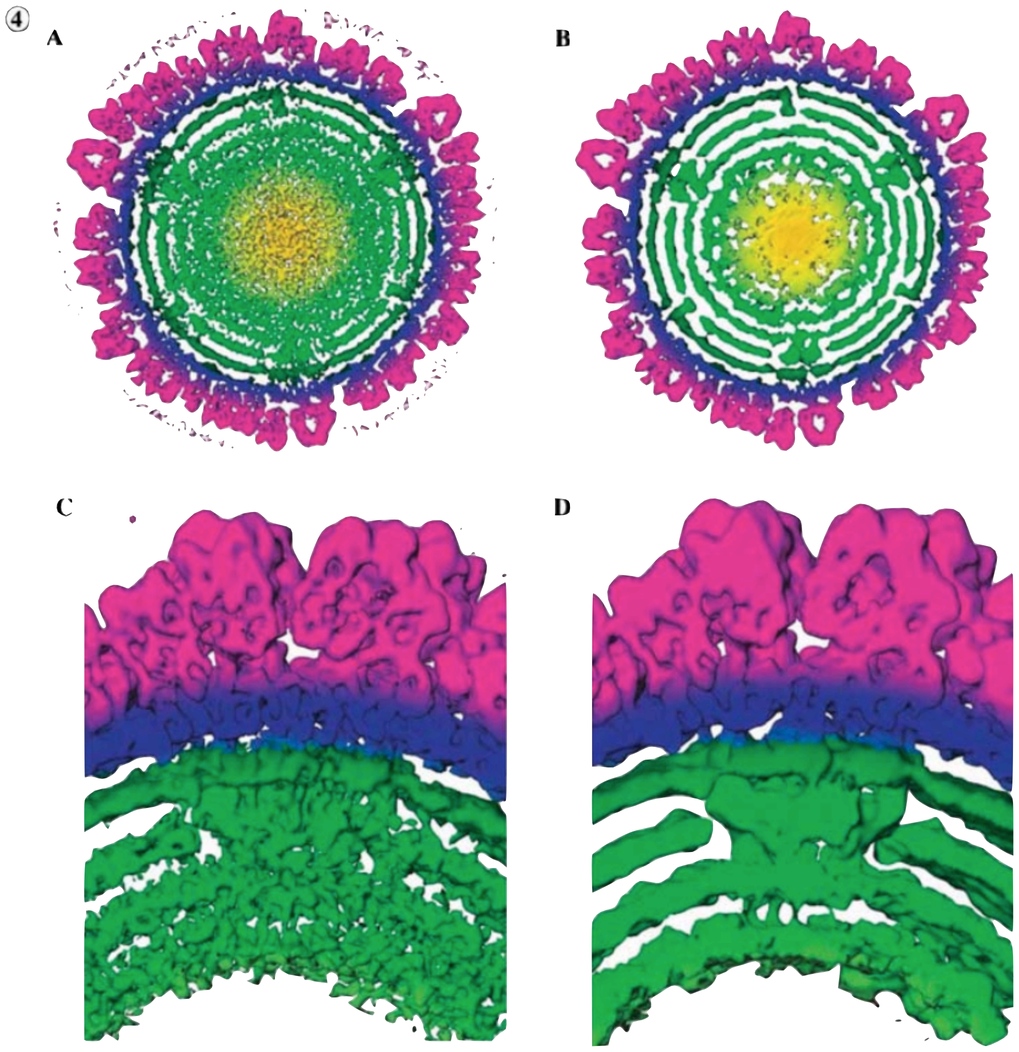
Wen Jiang,[a] Matthew L. Baker,[a] Qiu Wu,[b] Chandrajit Bajaj,[b] and Wah Chiu[a,*]

[a] *Verna and Marrs McLean Department of Biochemistry and Molecular Biology, Baylor College of Medicine, One Baylor Plaza, Houston, TX 77030, USA*

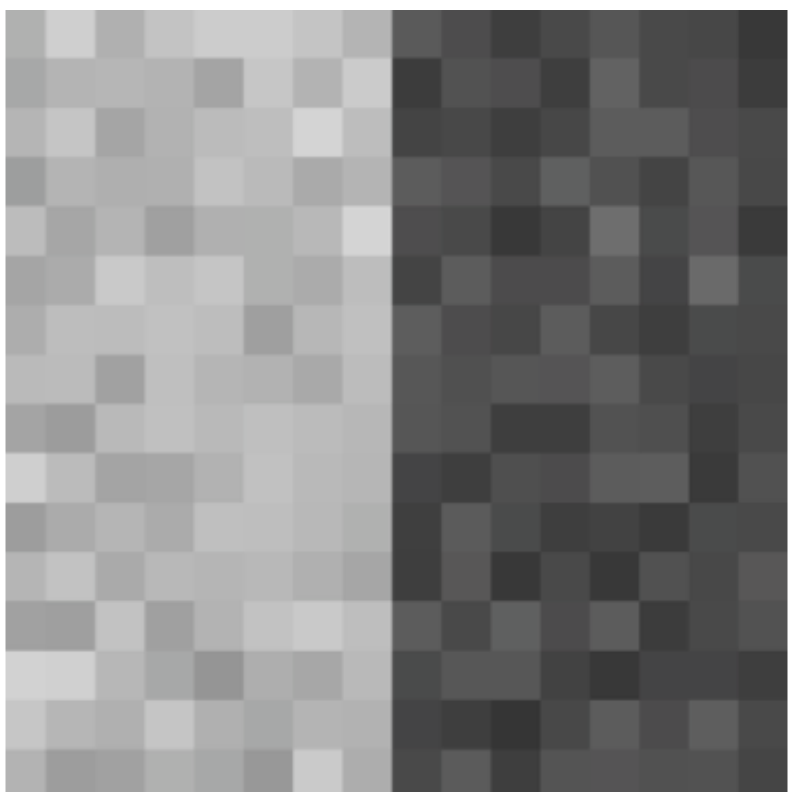
[b] *Institute for Computational Engineering and Sciences, University of Texas at Austin, Austin, TX 78712, USA*

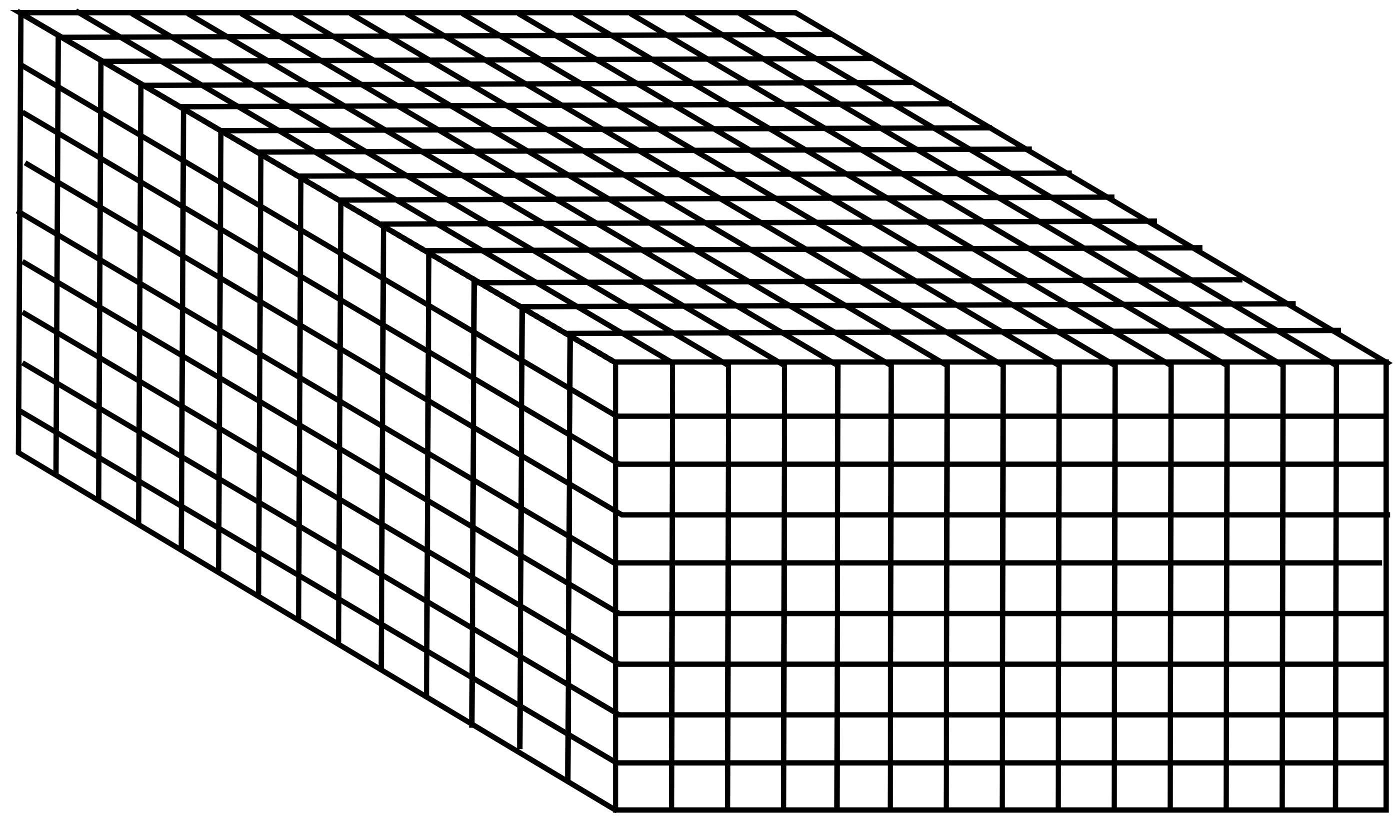
Received 20 August 2003, and in revised form 17 September 2003



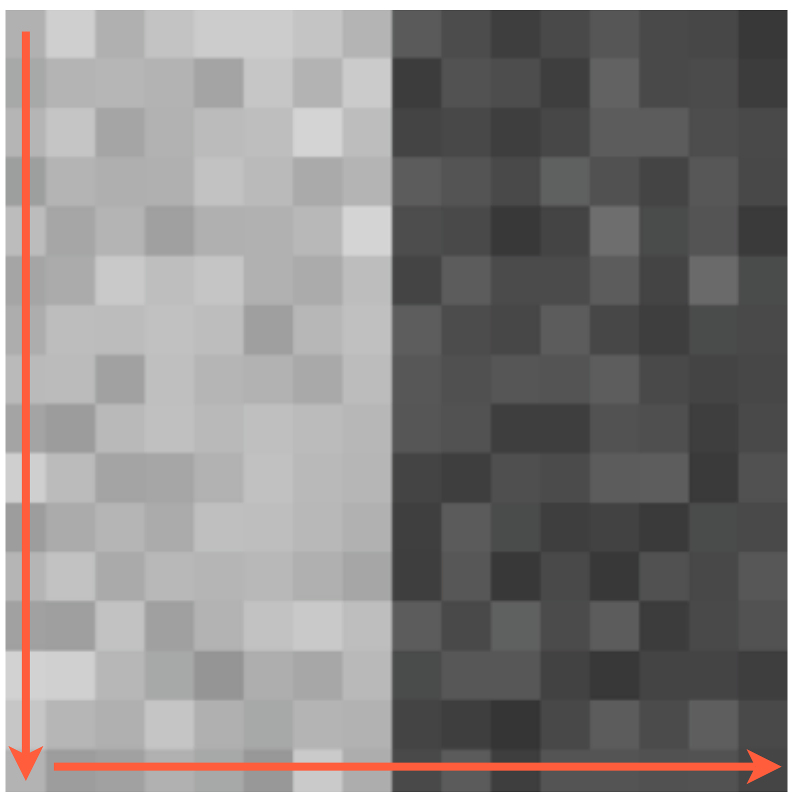


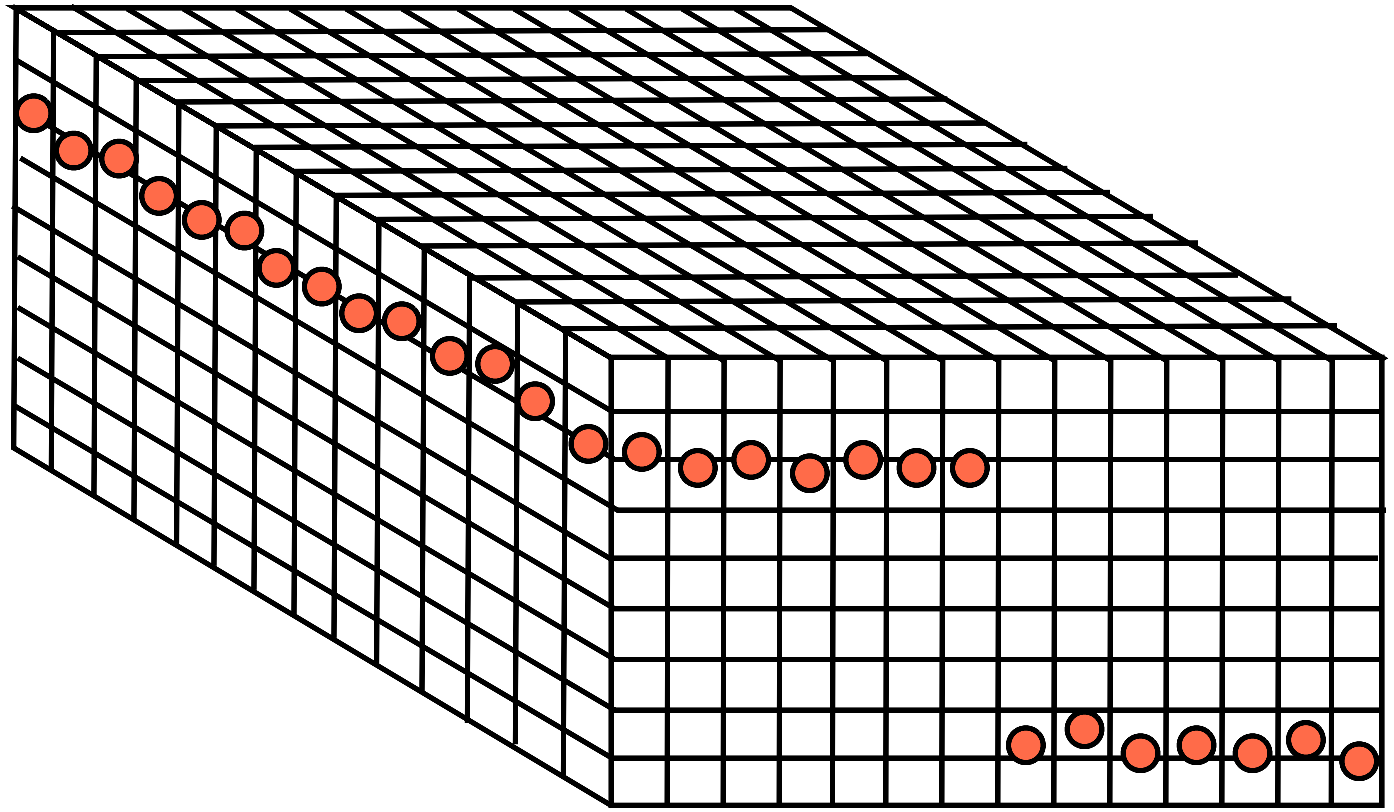
Bilateral Filter



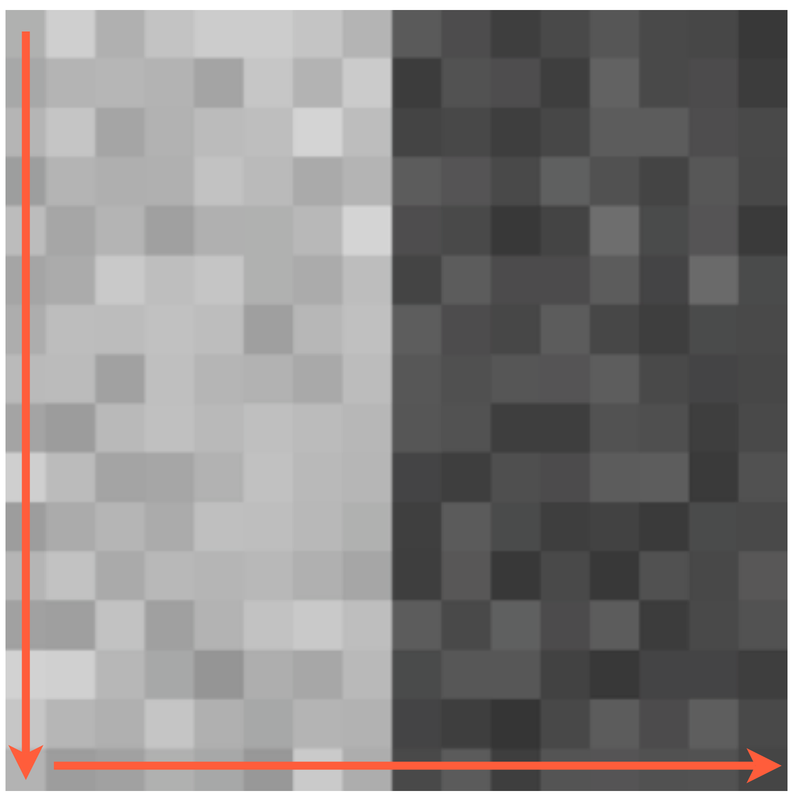


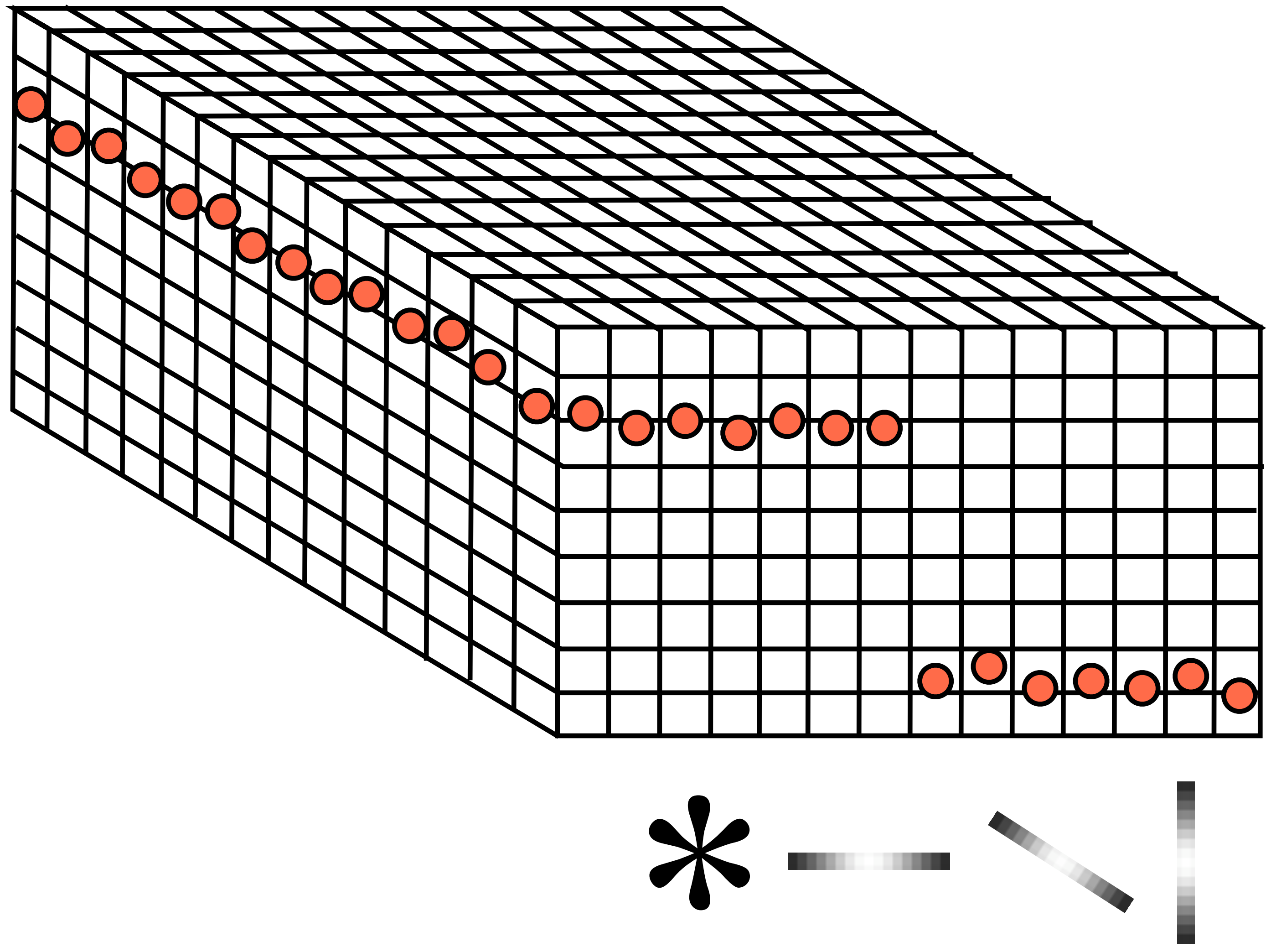
Bilateral Filter



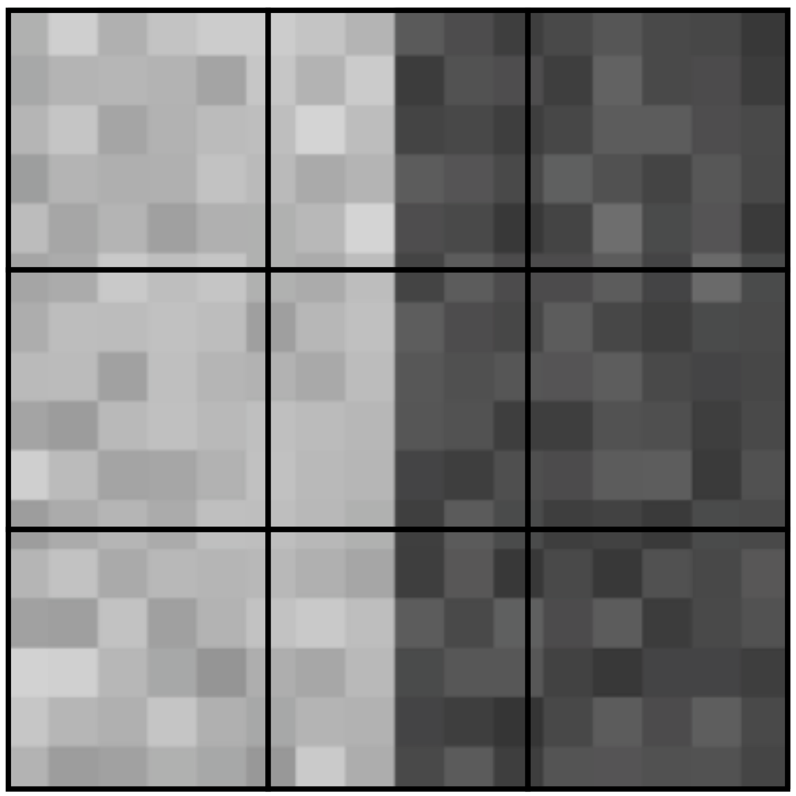


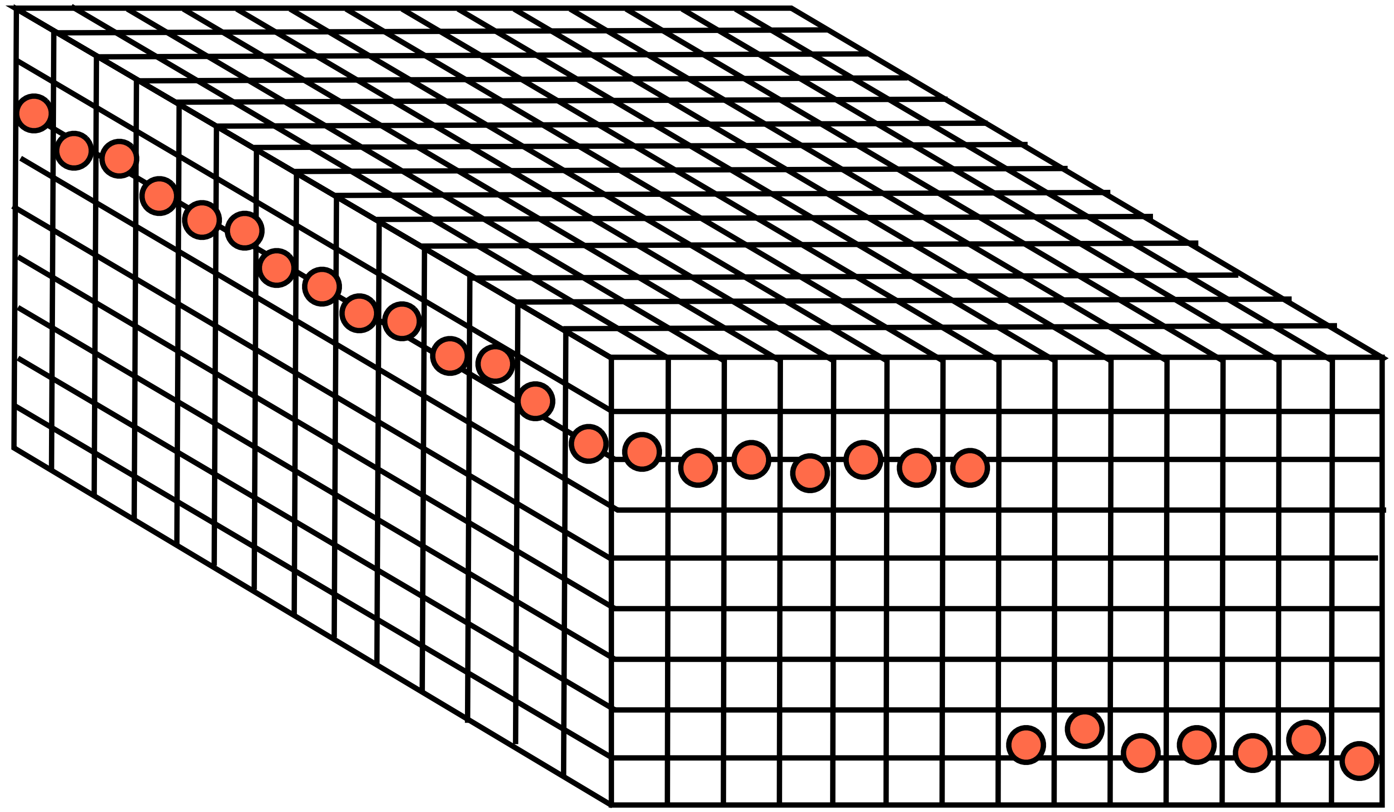
Bilateral Filter



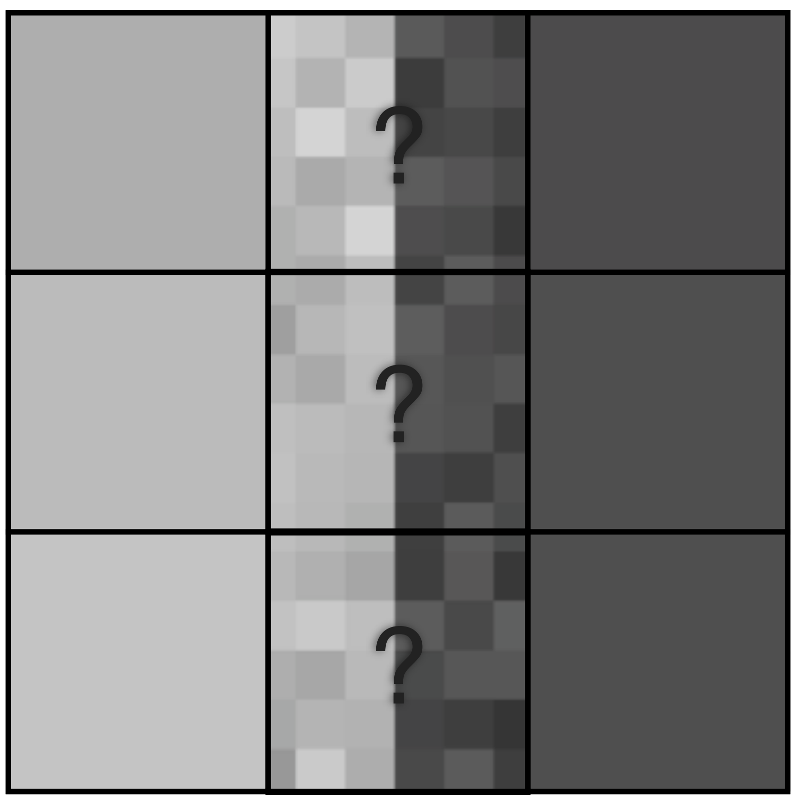


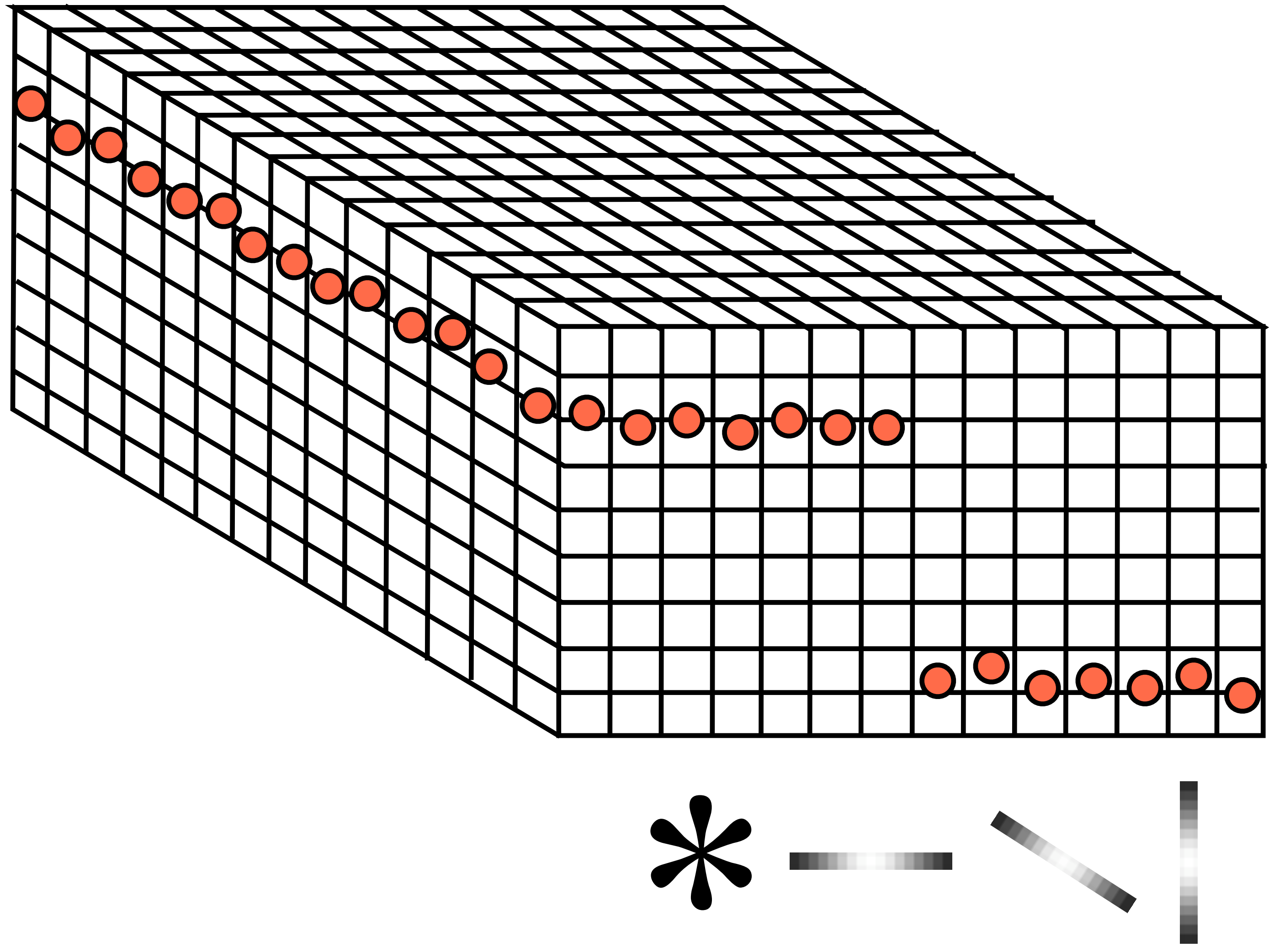
Bilateral Filter



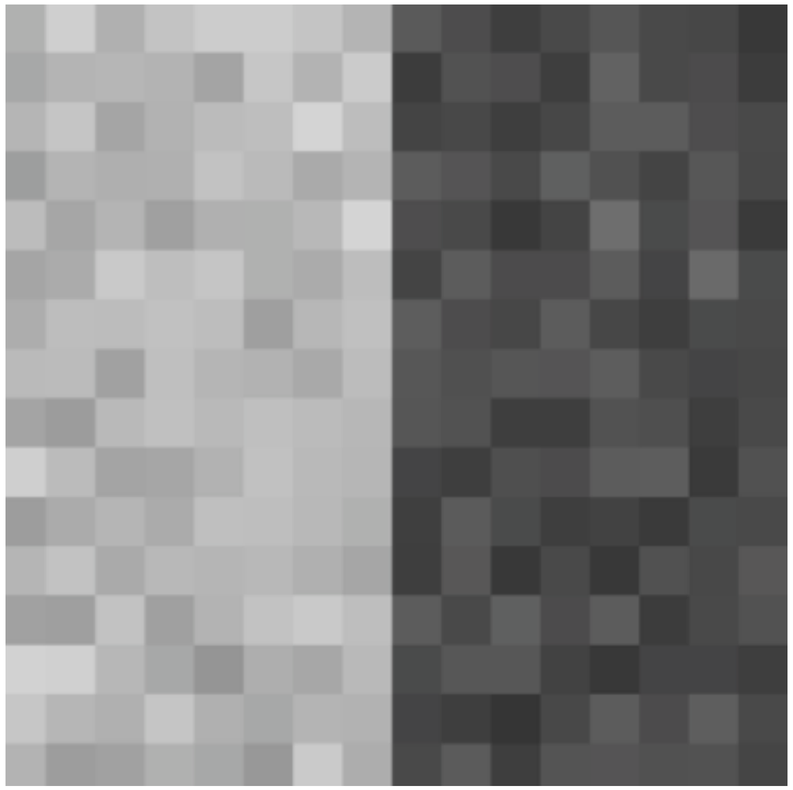


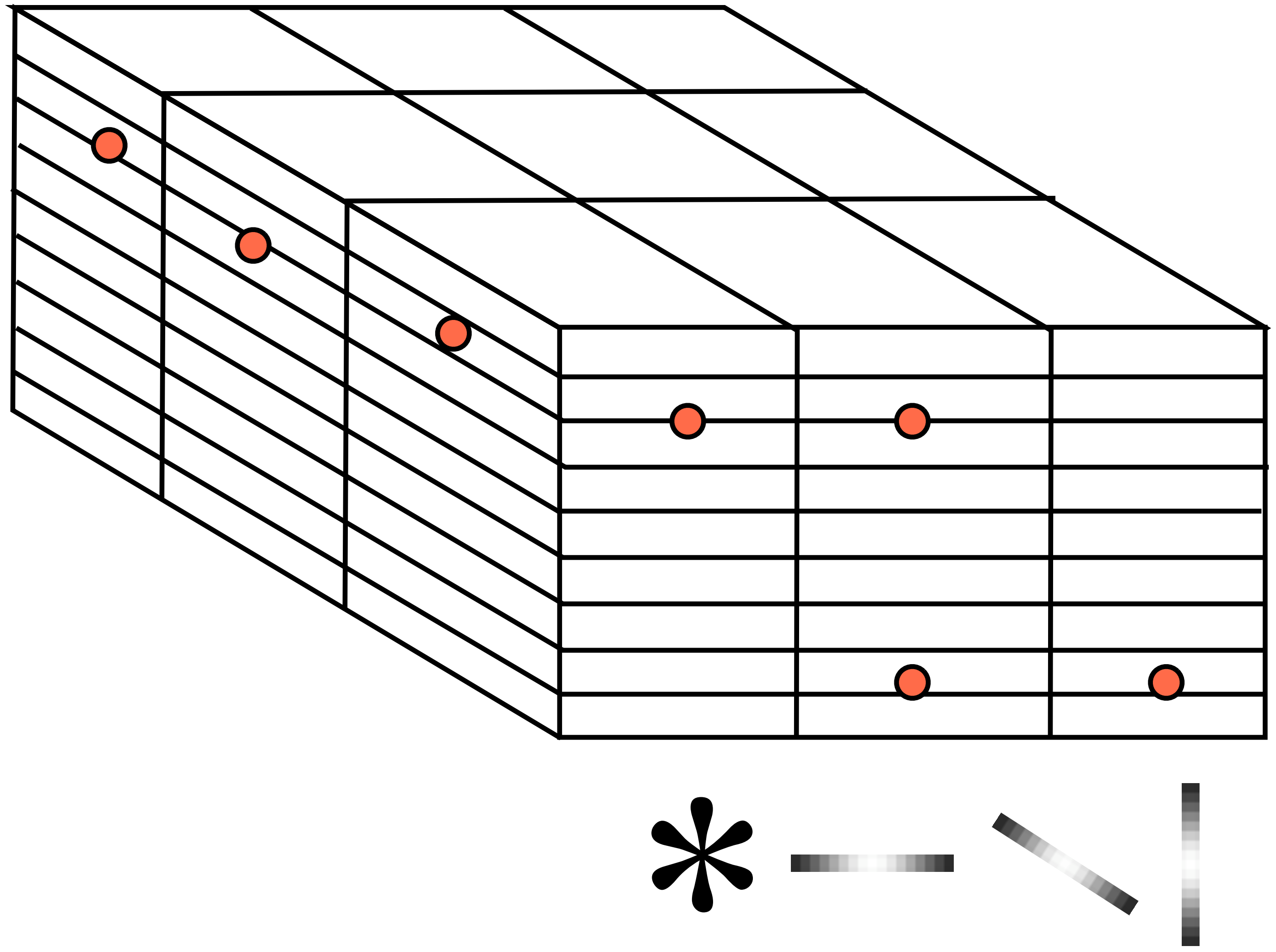
Bilateral Filter



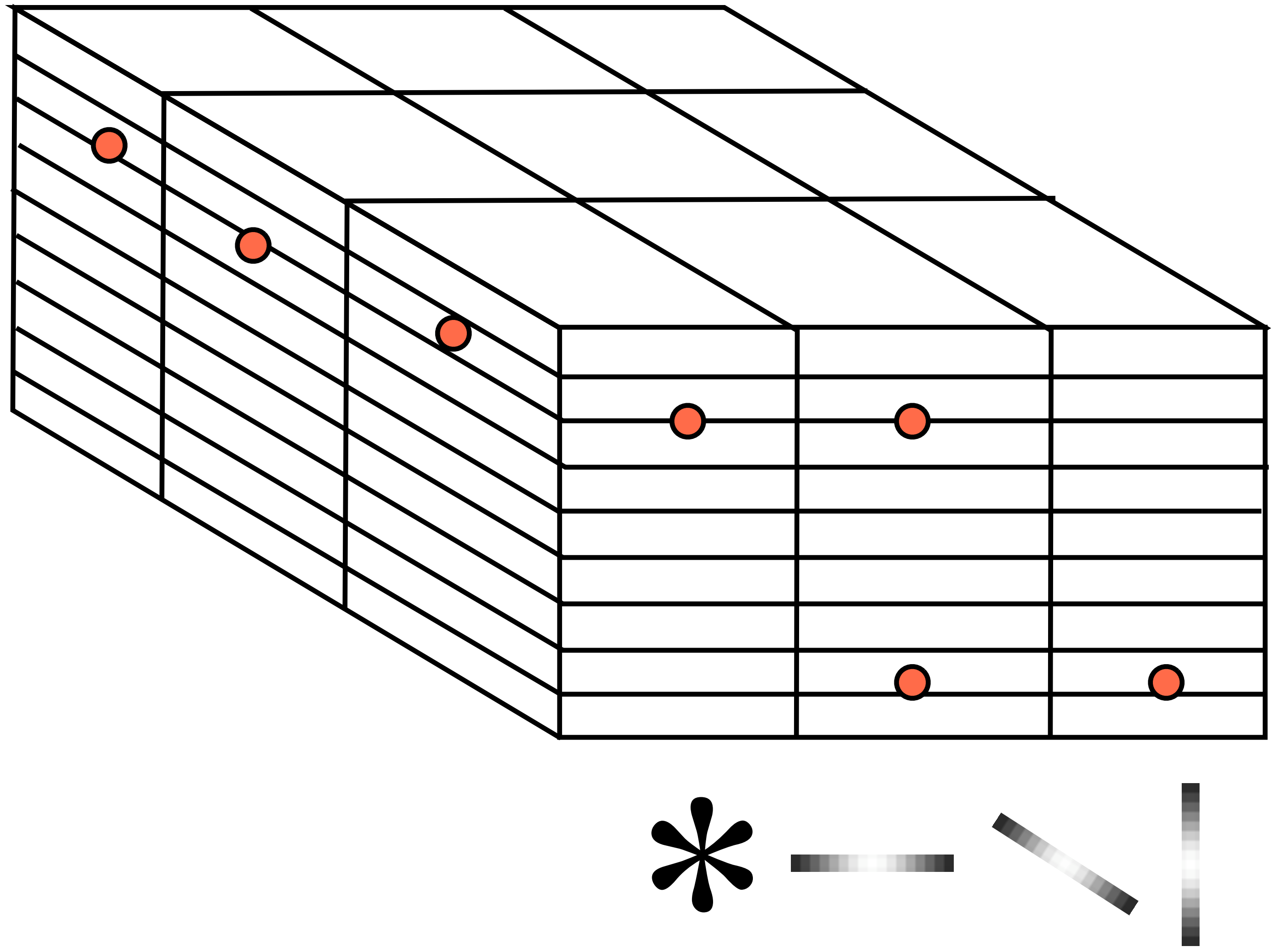


Bilateral Filter





Bilateral Filter



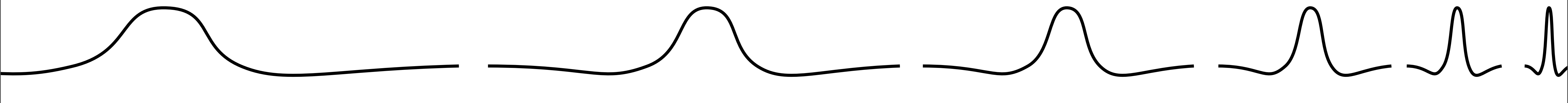
Bilateral Filter

- Brute-force Implementation
  - full $O(n^2)$ ---> truncated $O(n\sigma^2)$
- Box Kernel [1]
  - $O(n\lg\sigma)$ very fast, repeat for accuracy
- 3D Kernel [2]
  - $O(n+(n/\sigma^2)(r/\sigma))$ fast, accurate
  - GPU implementation [3] (very fast)

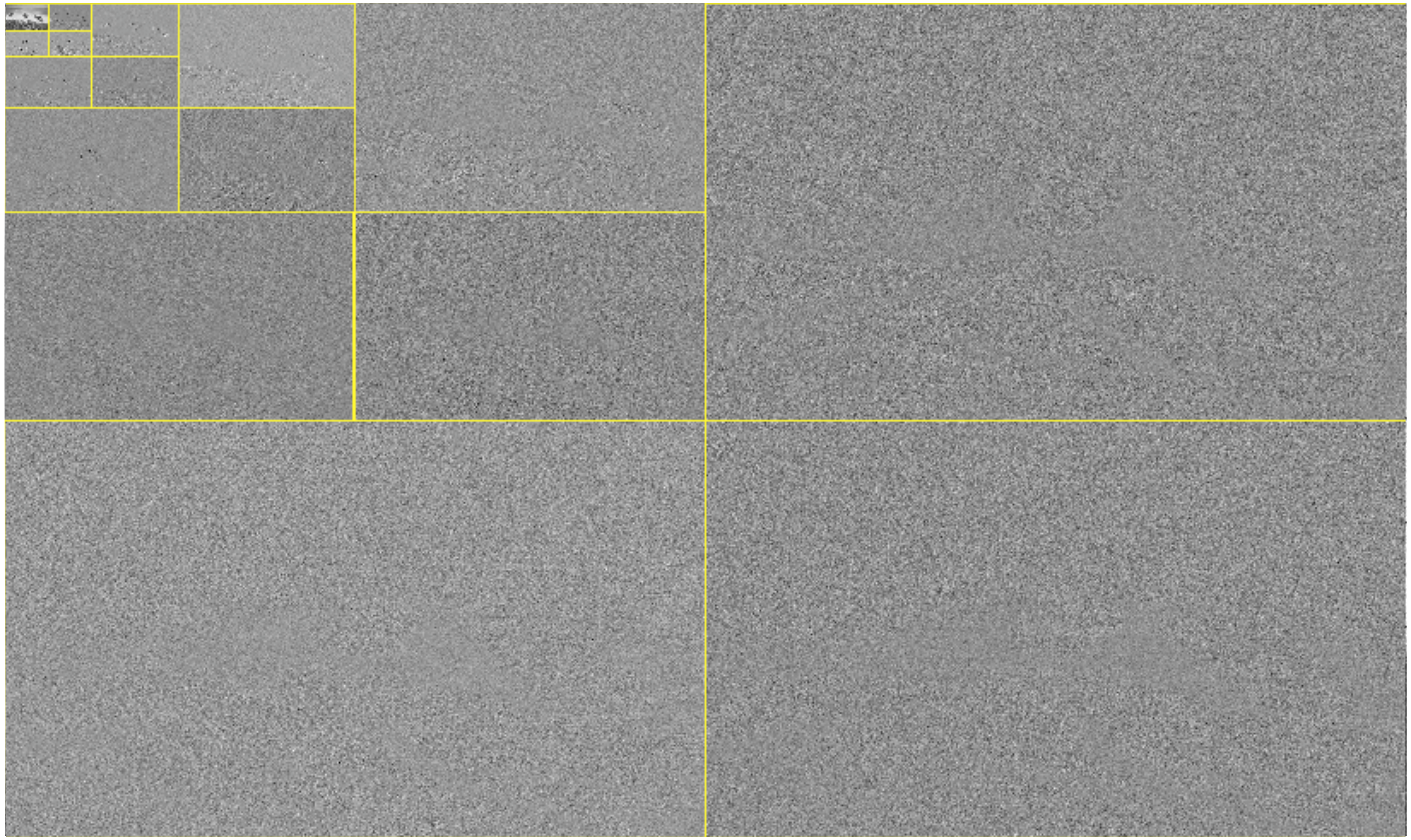
[1] Fast median and bilateral filtering, Ben Weiss
[2] A Fast Approximation of the Bilateral Filter using a Signal Processing Approach, Sylvain Paris and Frédo Durand
Code and paper: http://people.csail.mit.edu/sparis/bf/
[3] Real-time Edge-Aware Processing with the Bilateral Grid, Jiawen Chen, Sylvain Paris, Frédo Durand
Code and paper: http://groups.csail.mit.edu/graphics/bilagrid/

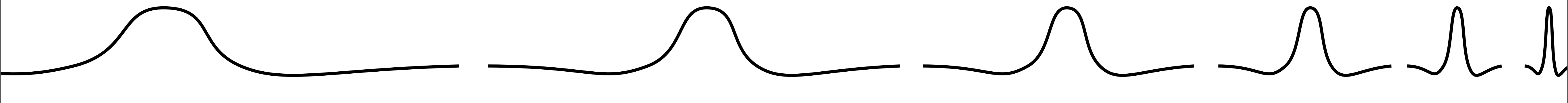
Wavelets

Wavelets

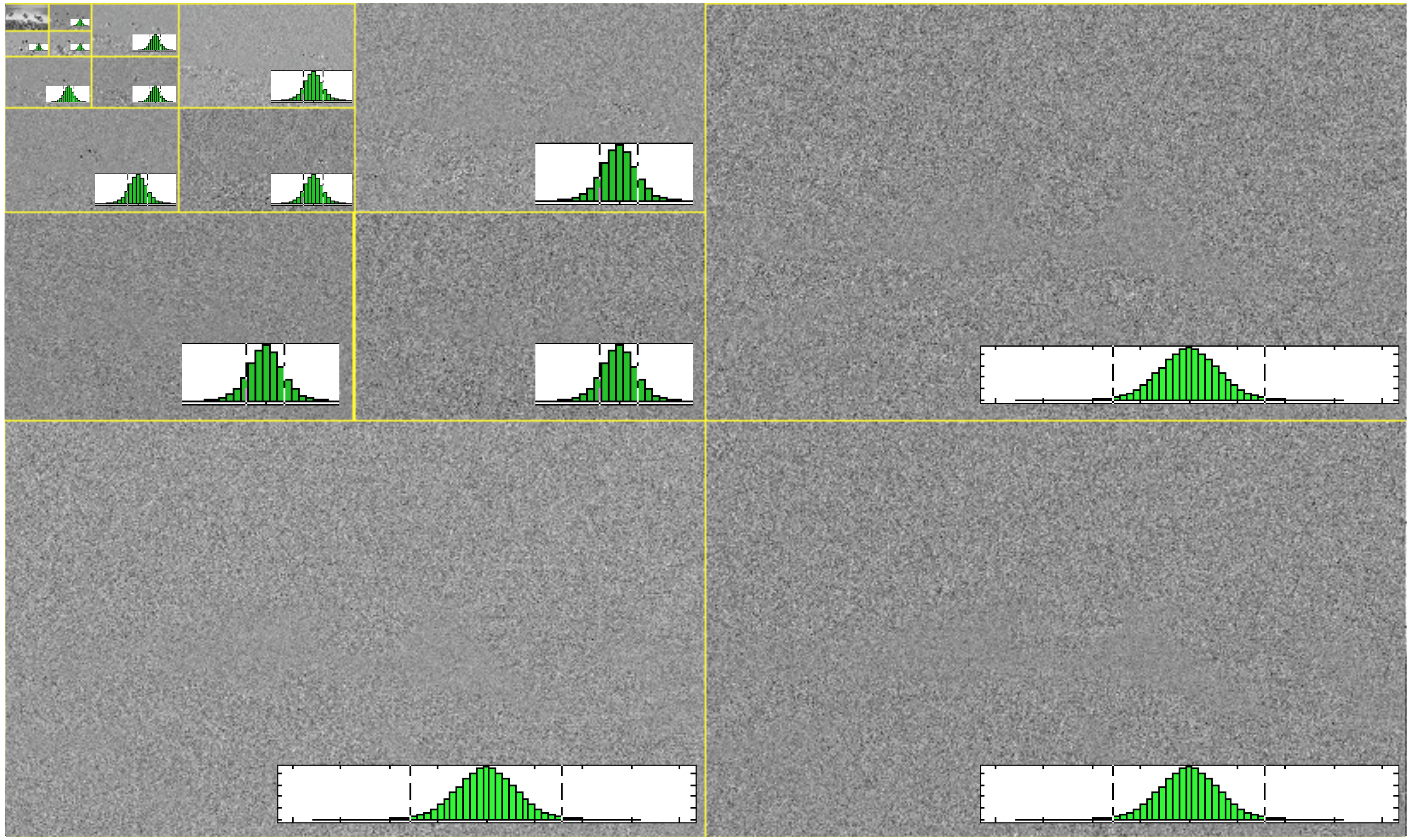


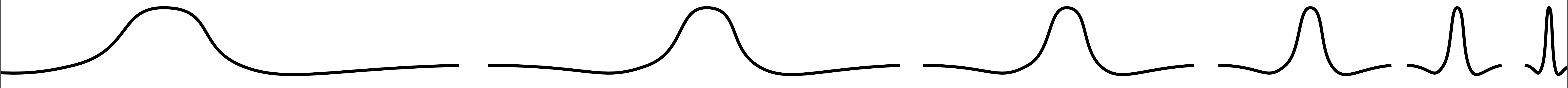
Wavelets



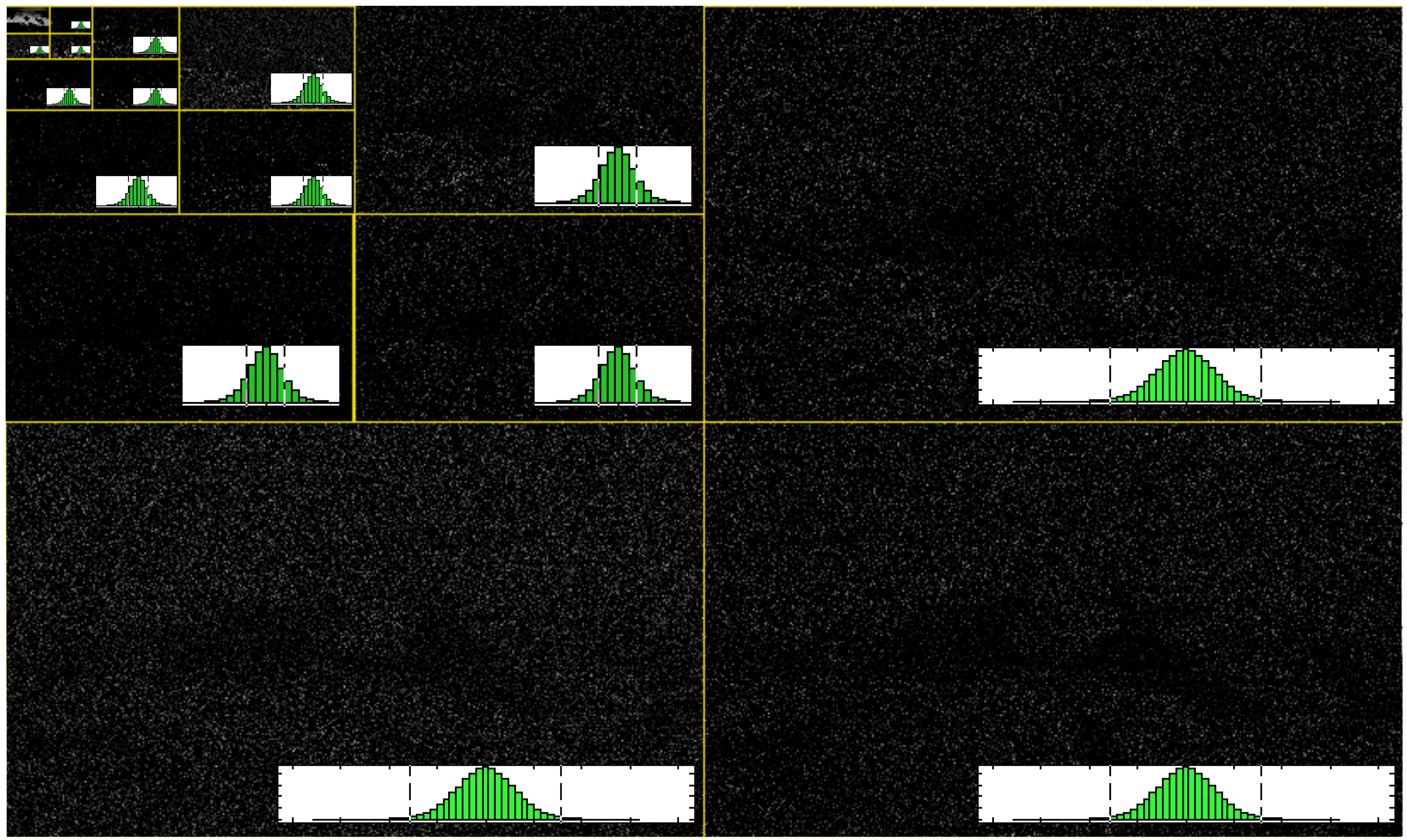


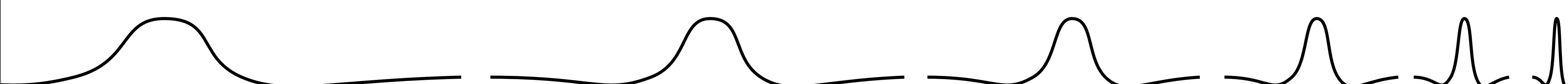
Wavelets



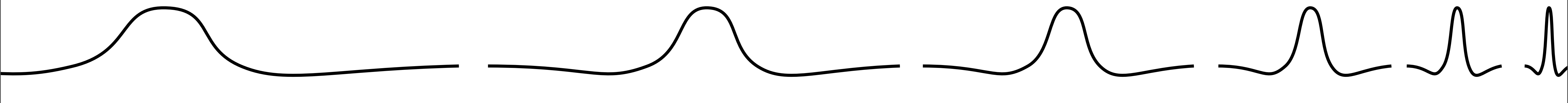


Wavelets





Wavelets



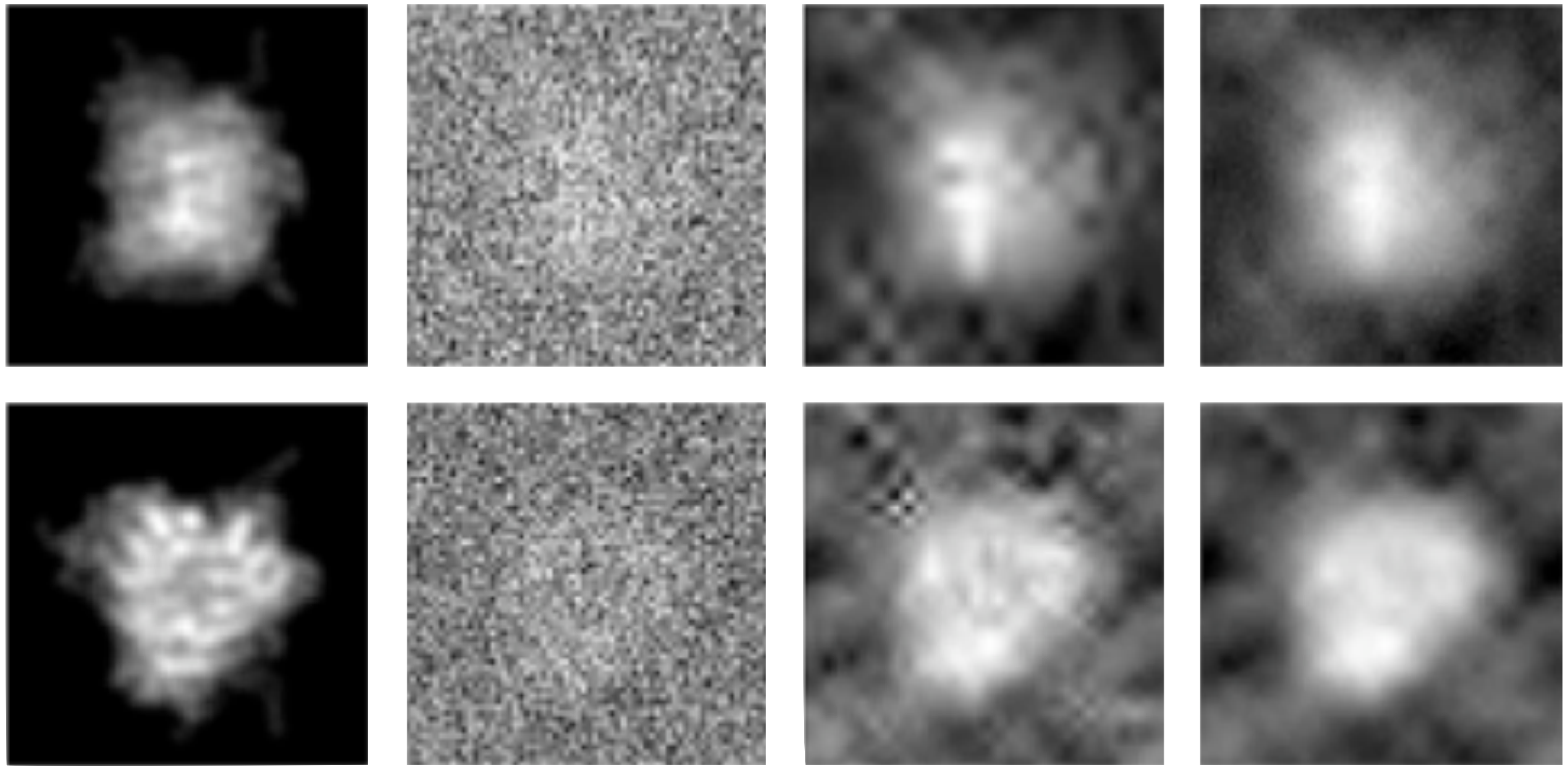
Wavelet Denoising

Improved Bayesian image denoising based on wavelets with applications to electron microscopy

C.O.S. Sorzano[a, b, *], E. Ortiz[a], M. López[c], J. Rodrigo[d]

[a]Dept. Sistemas Electrónicos y de Telecomunicación, Escuela Politécnica Superior, Univ. San Pablo-CEU, Urb. Montepríncipe s/n, Boadilla del Monte, 28668 Madrid, Spain
[b]Biocomputing Unit, National Center of Biotechnology (CSIC), Campus Univ. Autónoma s/n, 28049 Madrid, Spain
[c]Dept. Matemática e Informática Aplicadas a la Ingeniería Civil, E.T.S. Ingenieros de Caminos, Univ. Politécnicade Madrid, Ciudad Universitaria s/n, 28040 Madrid, Spain
[d]Dept. Matemática Aplicada, E.T.S. Ingeniería, Univ. Pontificia Comillas, c/Alberto Aguilera, 23, 28015 Madrid, Spain

Received 14 June 2004; received in revised form 1 December 2005; accepted 1 December 2005

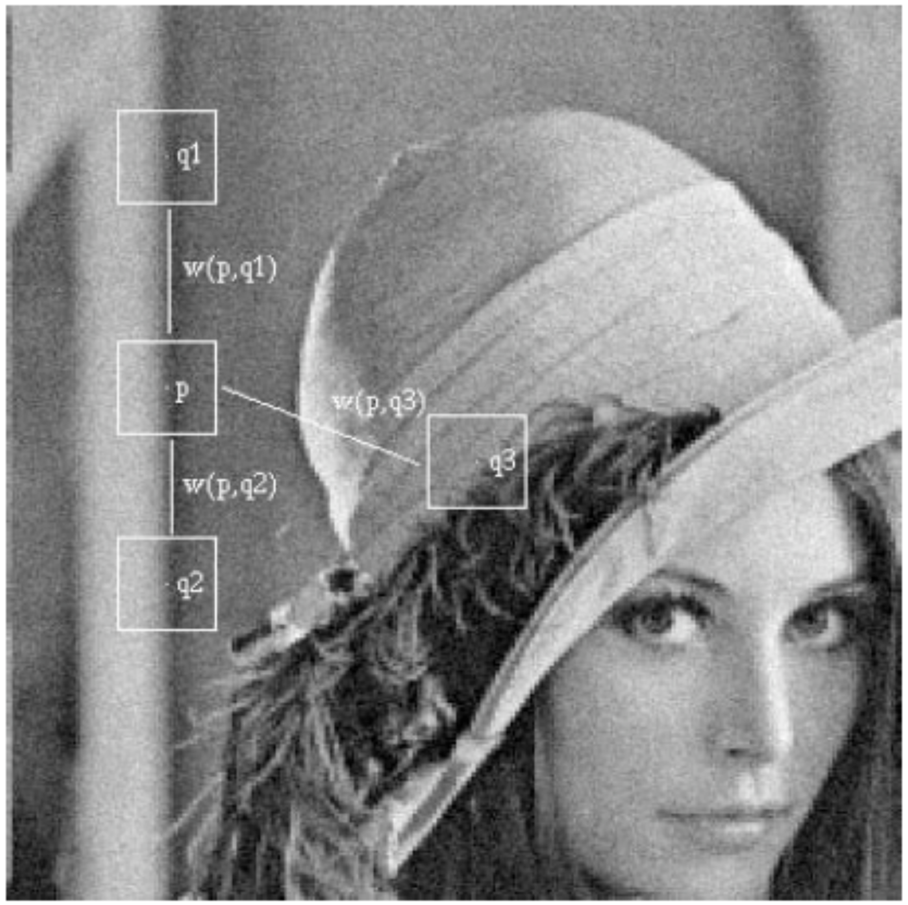


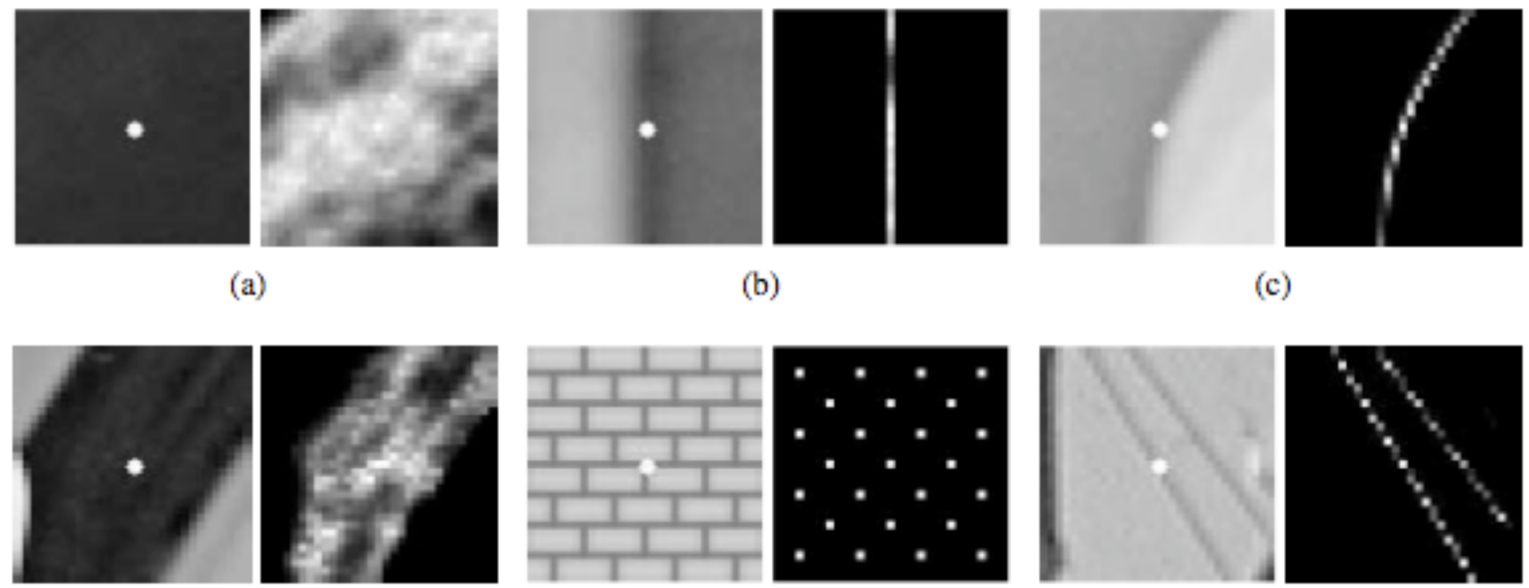
Wavelet Denoising

- Decomposition and Recomposition in O(n)
- Choice of wavelet functions
- Choice of wavelet filter methods
  - Thresholding [1]
  - Wiener Filtering [2]
  - Bayesian Filtering [3]

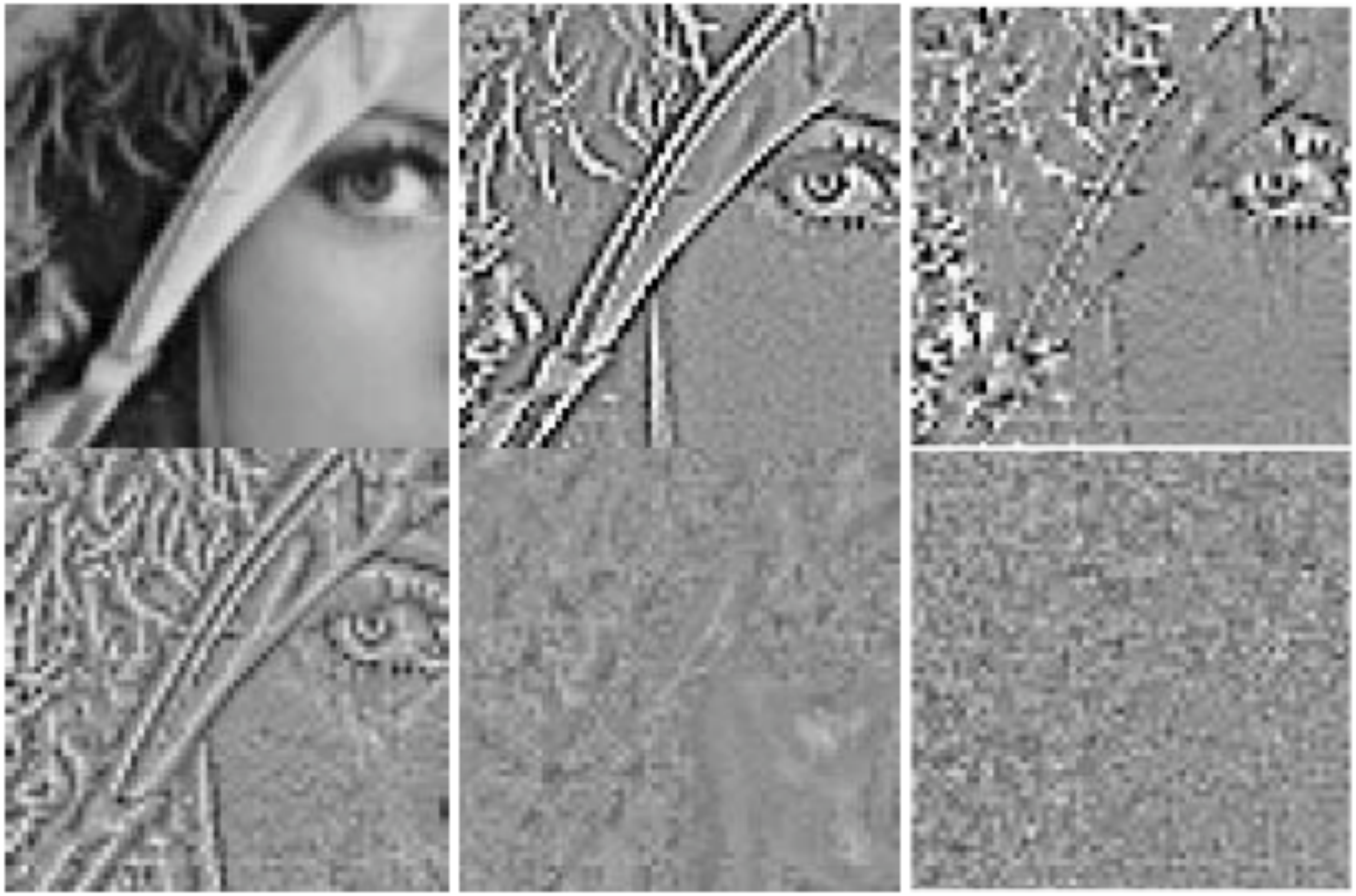
[1] Adaptive wavelet thresholding for image denoising and compression, Chang, S.G., Bin Yu, Vetterli, M.
[2] J.L. Starck, A. Bijaoui, Filtering and deconvolution by the wavelet transform, Signal Process. 35 (1994) 195–211
[3] J. Portilla, V. Strela, M.J. Wainwright, E.P. Simoncelli, Denoising using scale mixtures of Gaussians in the wavelet domain, IEEE Trans. Image Process. 12 (2003) 1338–1351

Non-Local Means Filter





Method Noise



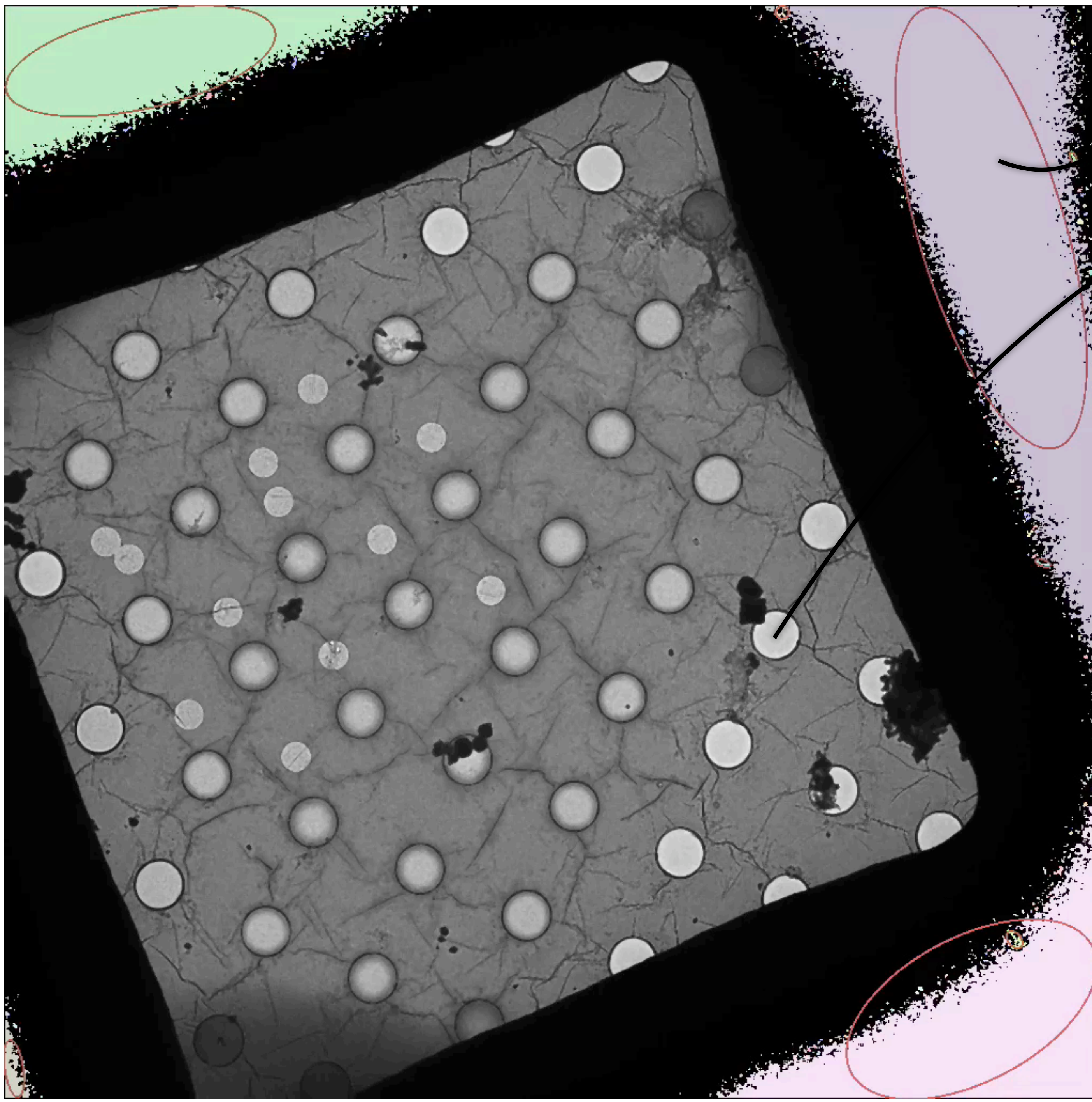
Denoising Overview

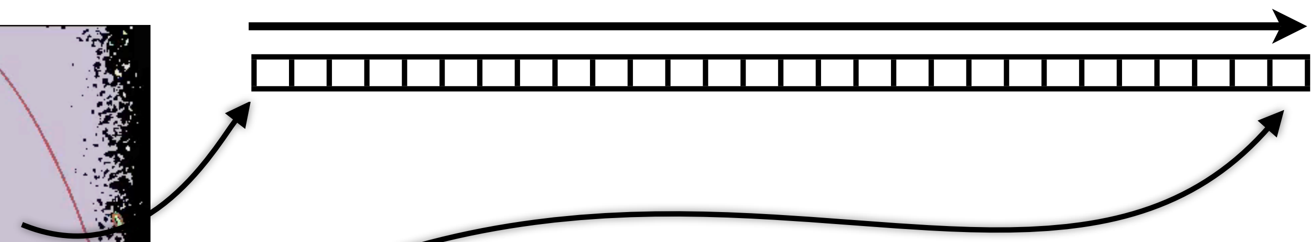
- Bilateral Filters
  - Very, very efficient (Fast CPU and GPU implementations)
  - Decent Noise Reduction
- Wavelet Filters
  - Very Efficient
  - Better Noise Reduction
- NL-Means
  - SLOW
  - State-of-the-art (on images, unknown for EM Images)

MSER
Maximally Stable Extremal Regions

MSER

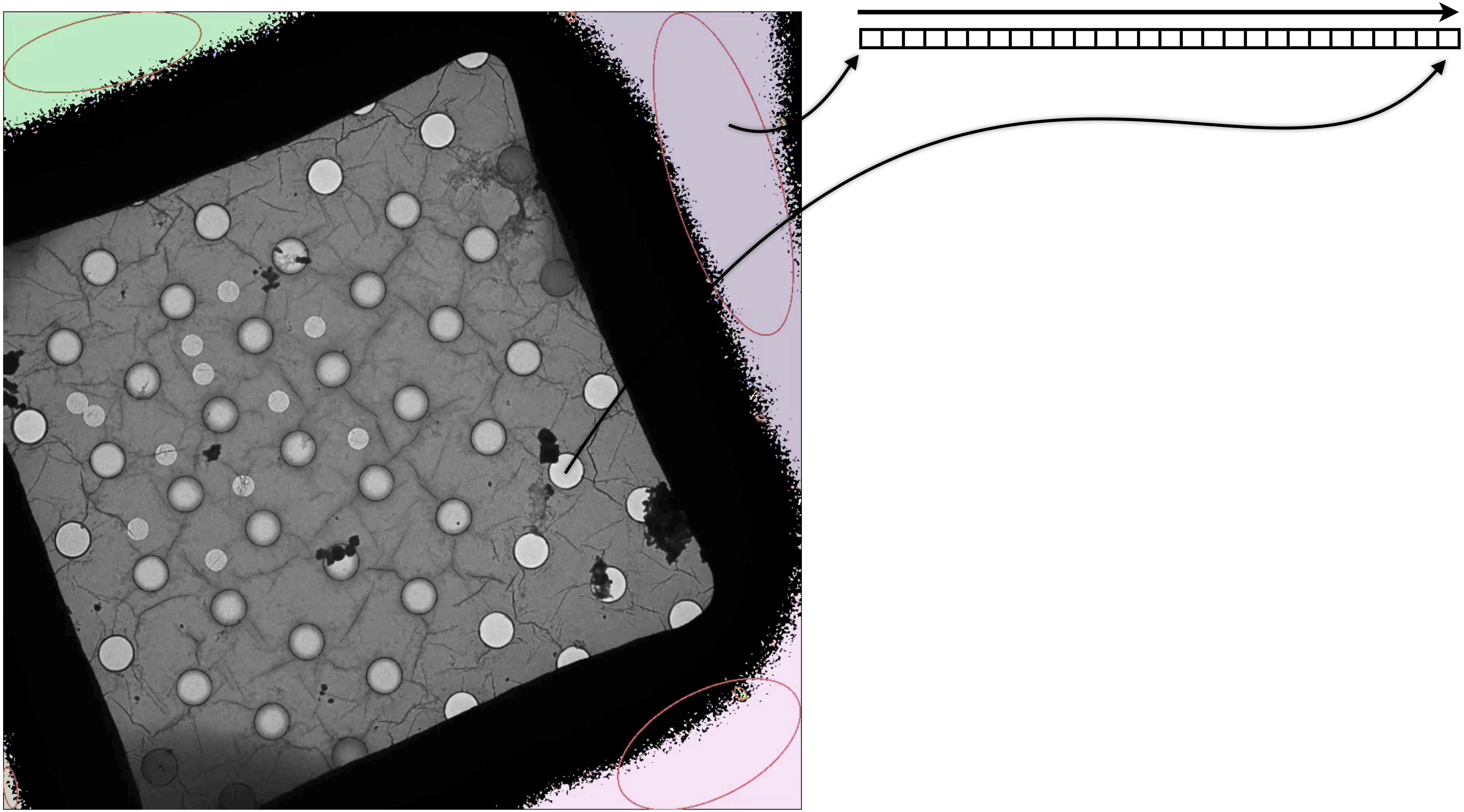
Maximally Stable Extremal Regions





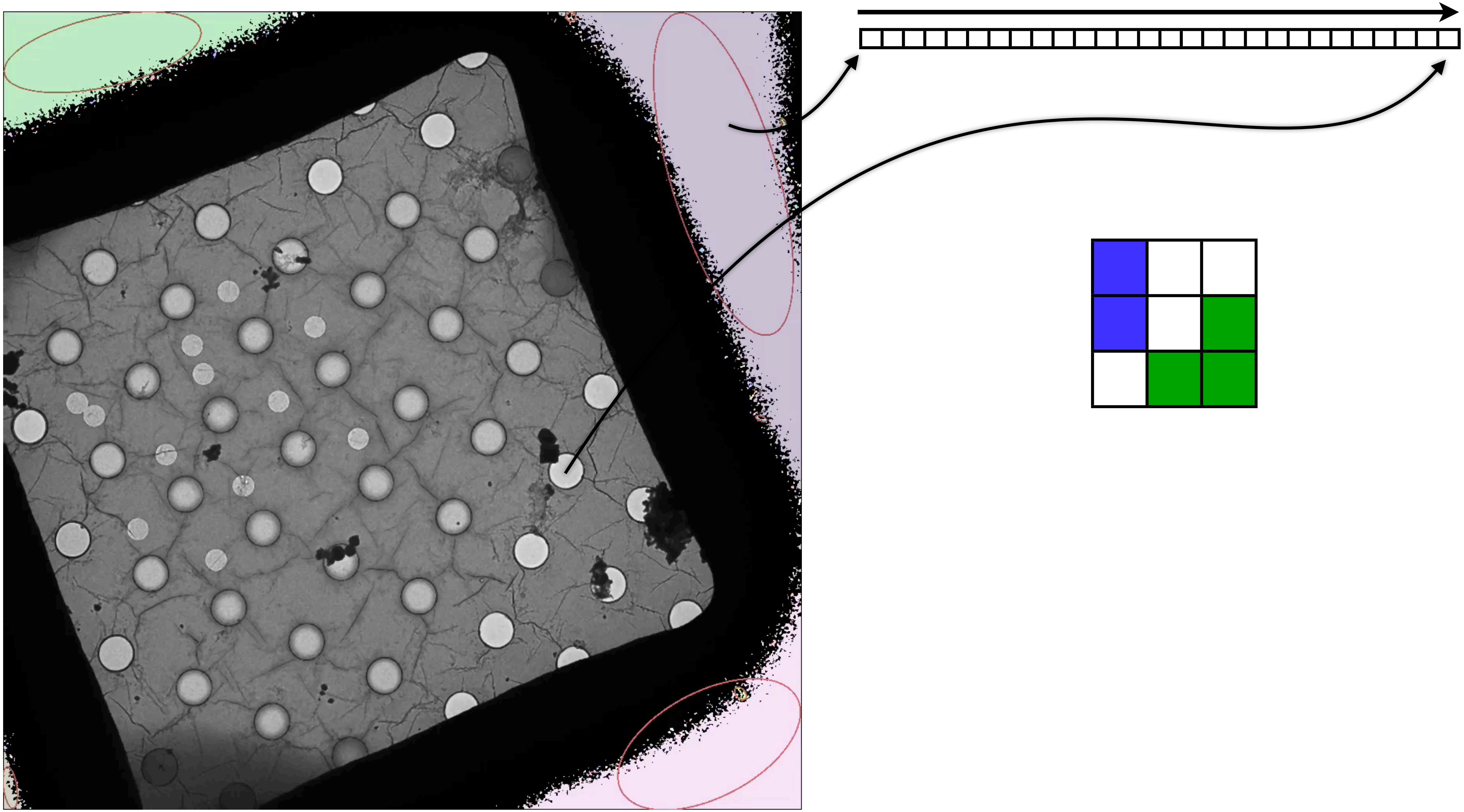
MSER

Maximally Stable Extremal Regions



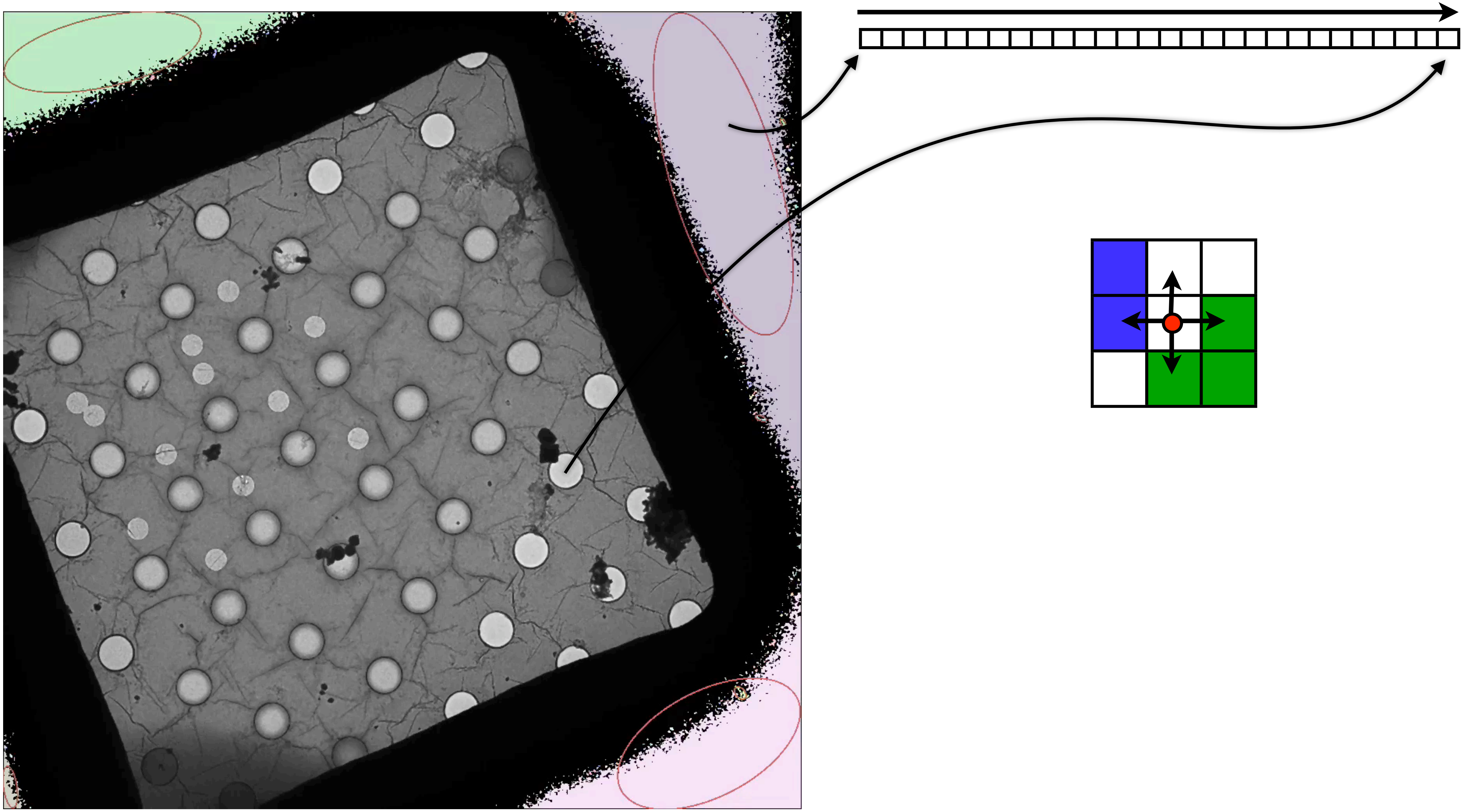
MSER

Maximally Stable Extremal Regions



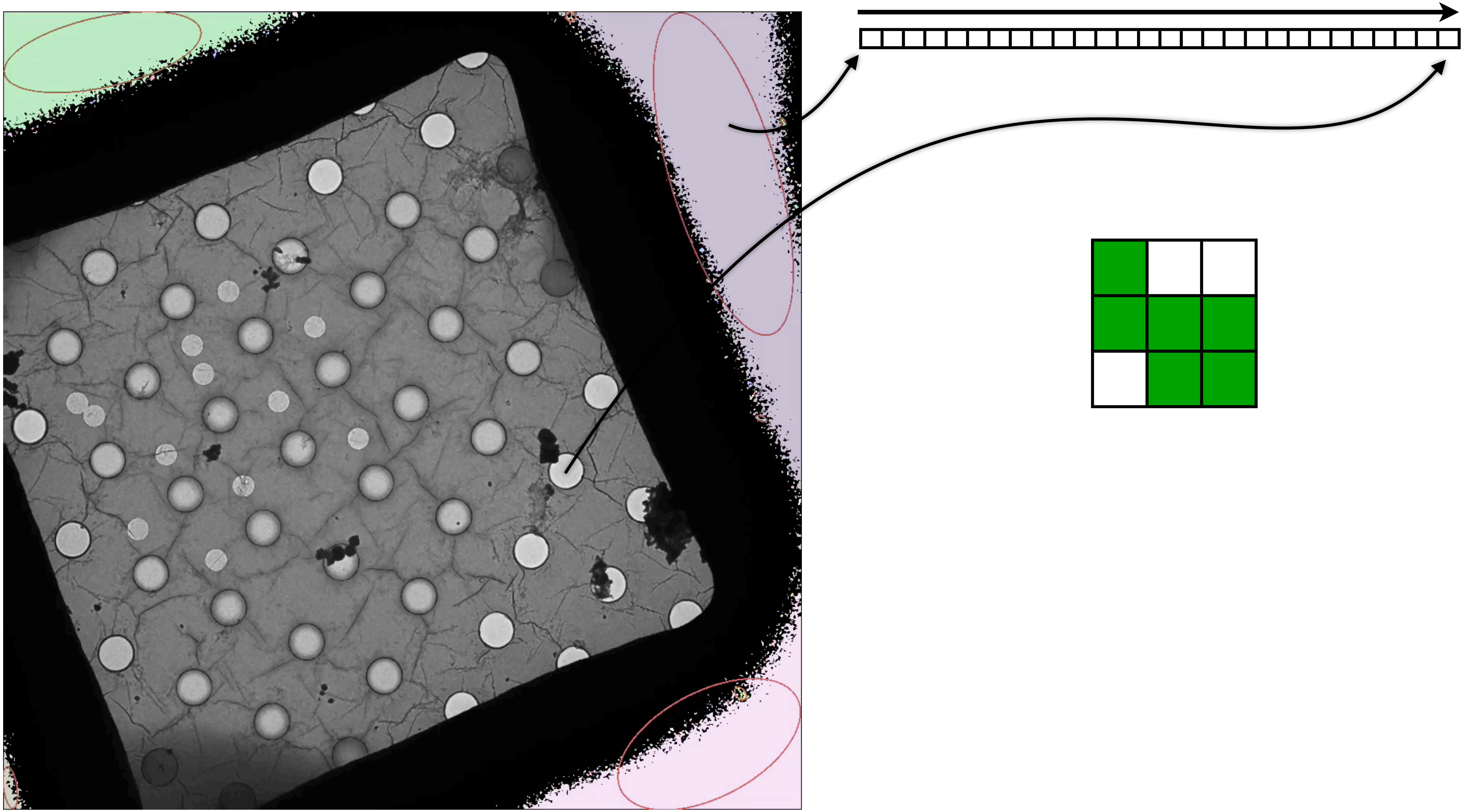
MSER

Maximally Stable Extremal Regions



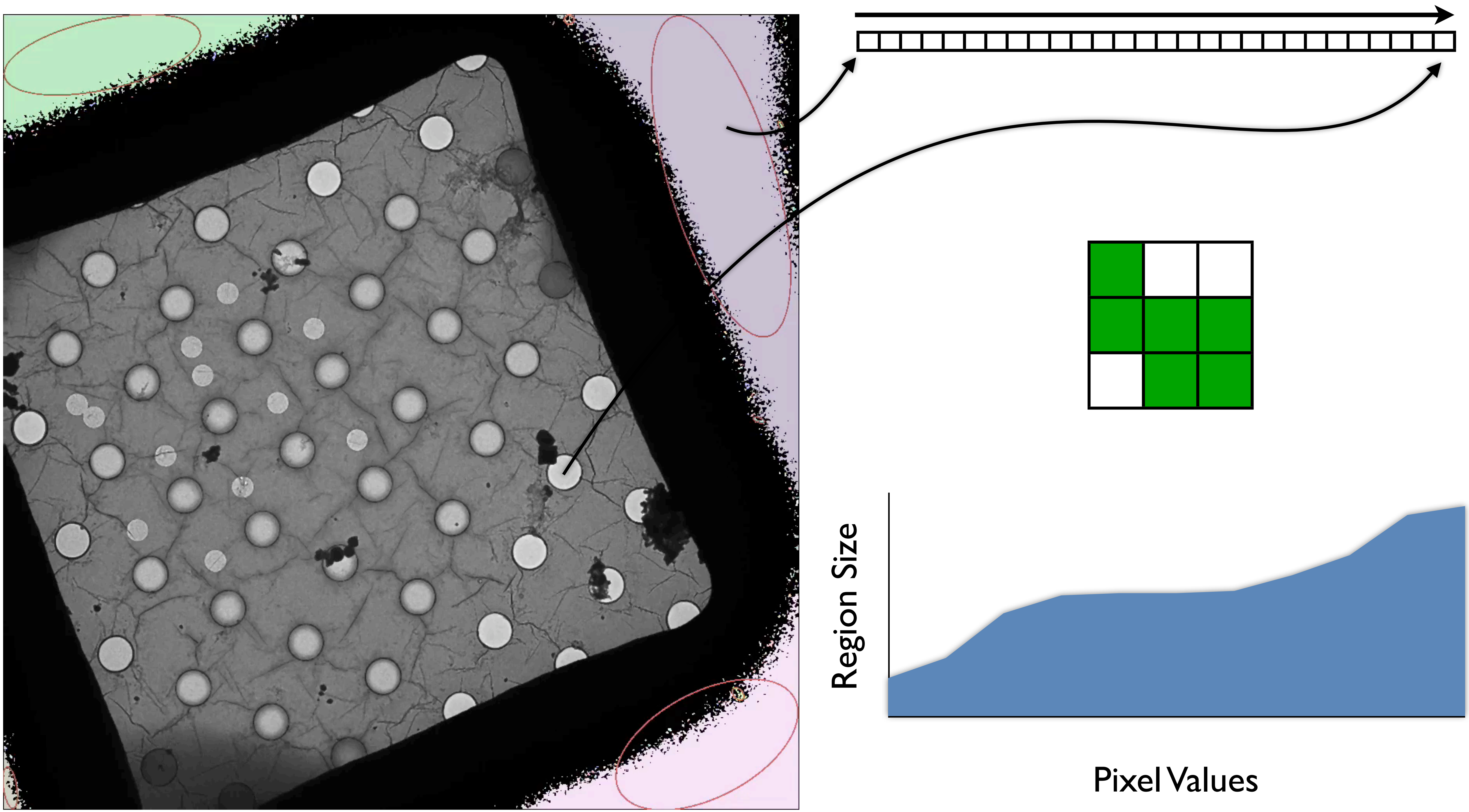
MSER

Maximally Stable Extremal Regions



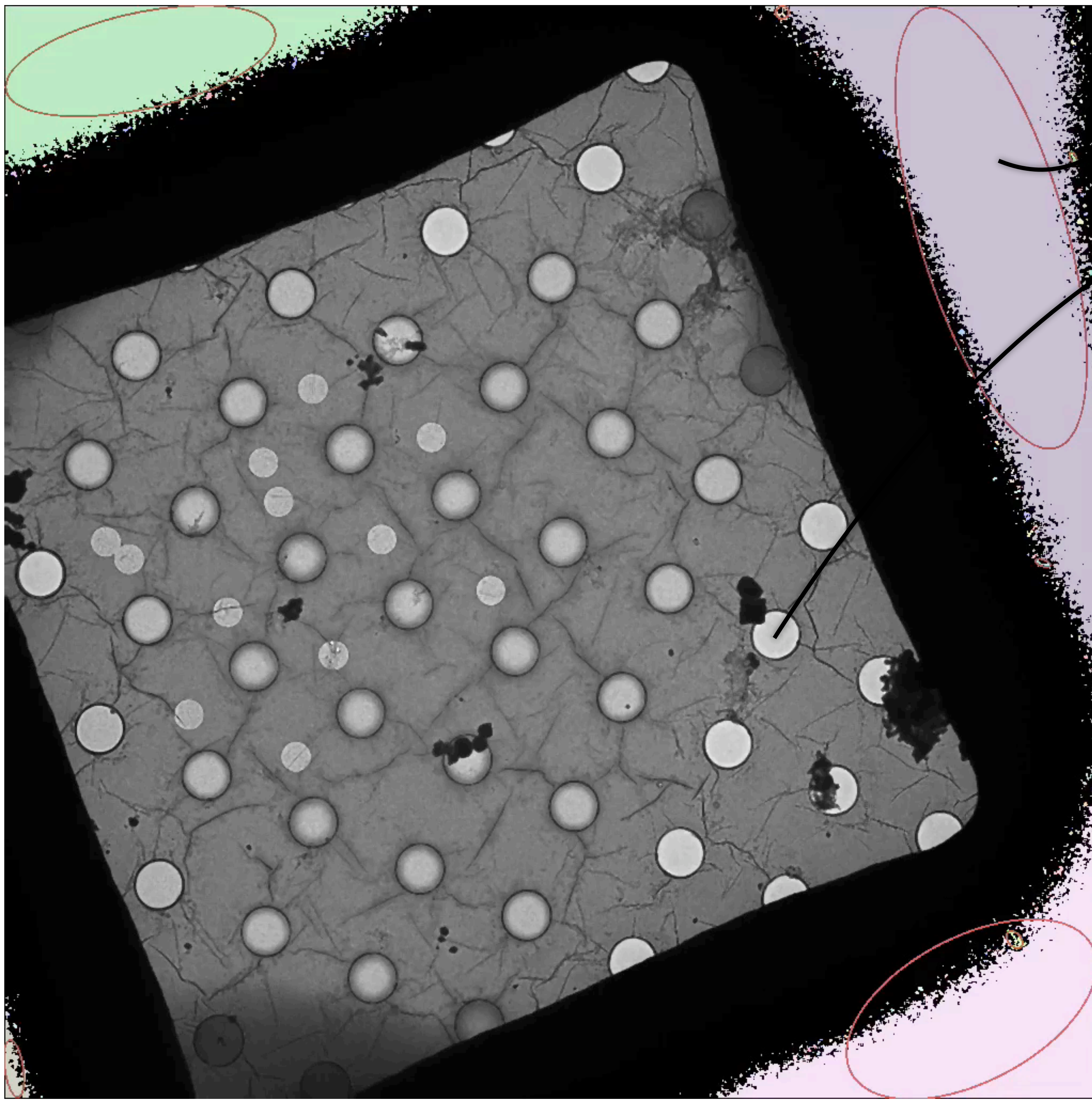
MSER

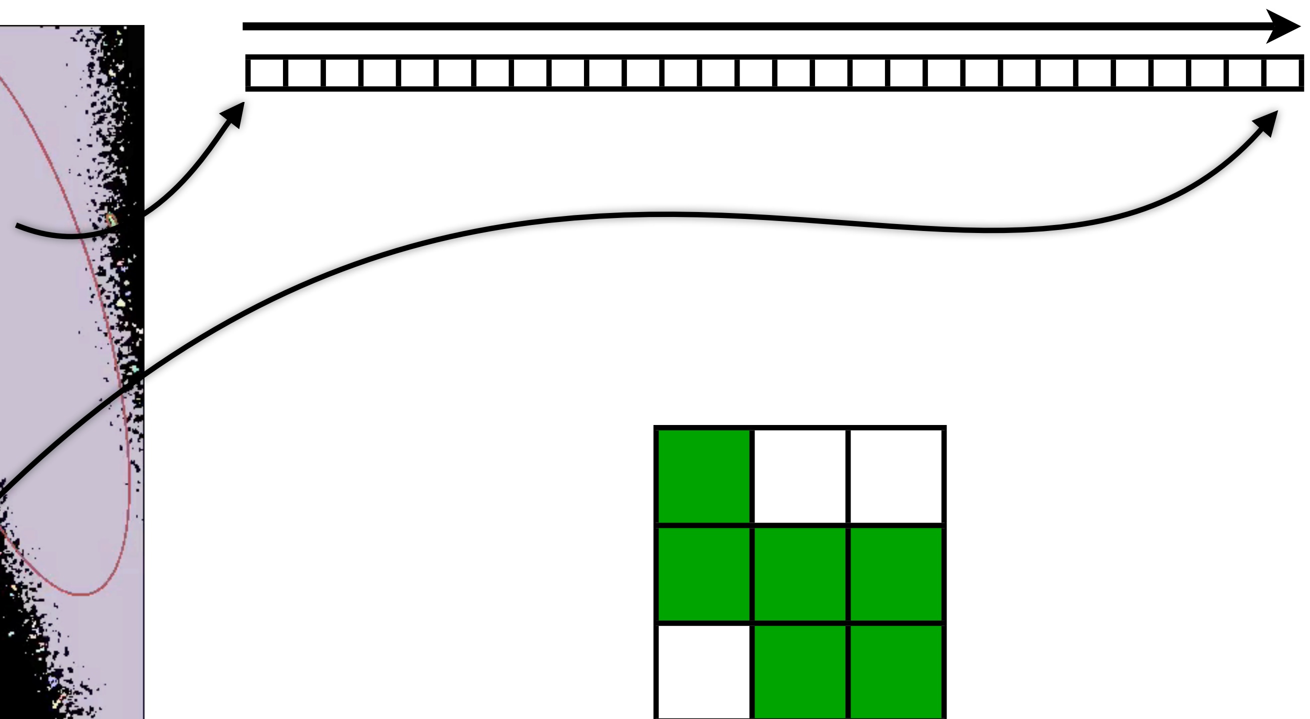
Maximally Stable Extremal Regions

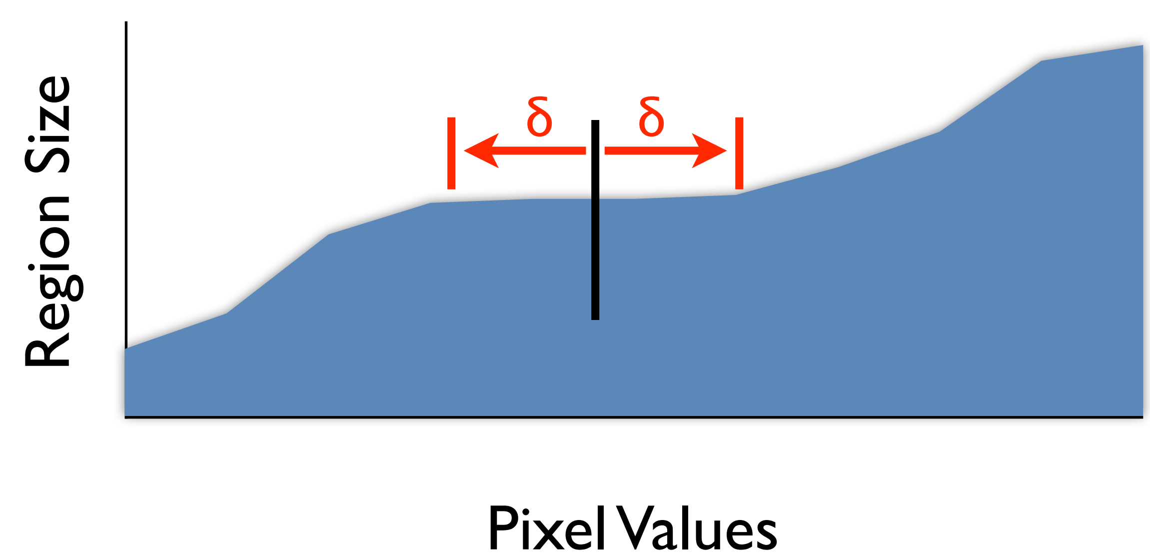


MSER

Maximally Stable Extremal Regions







MSER
Maximally Stable Extremal Regions

- Very efficient, $O(n\alpha(n))$, $\alpha(1X10^{500}) < 4$
- Easily extendable to higher dimensions
- Suitable for particle segmentation in conjunction with other image processing
- Suitable for 3D map segmentation

[1] Robust wide baseline stereo from maximally stable extremal regions. J. Matas, O. Chum, U. Martin, and T Pajdla. Proceedings of the British Machine Vision Conference, volume 1, pages 384-393, 2002.
Papers: http://cmp.felk.cvut.cz/~matas/
[2] An Implementation of Multi-Dimensional Maximally Stable Extremal Regions. Andrea Vedaldi
Code(C+Matlab) and PDF: http://vision.ucla.edu/~vedaldi/code/mser/mser.html

It's LoG!

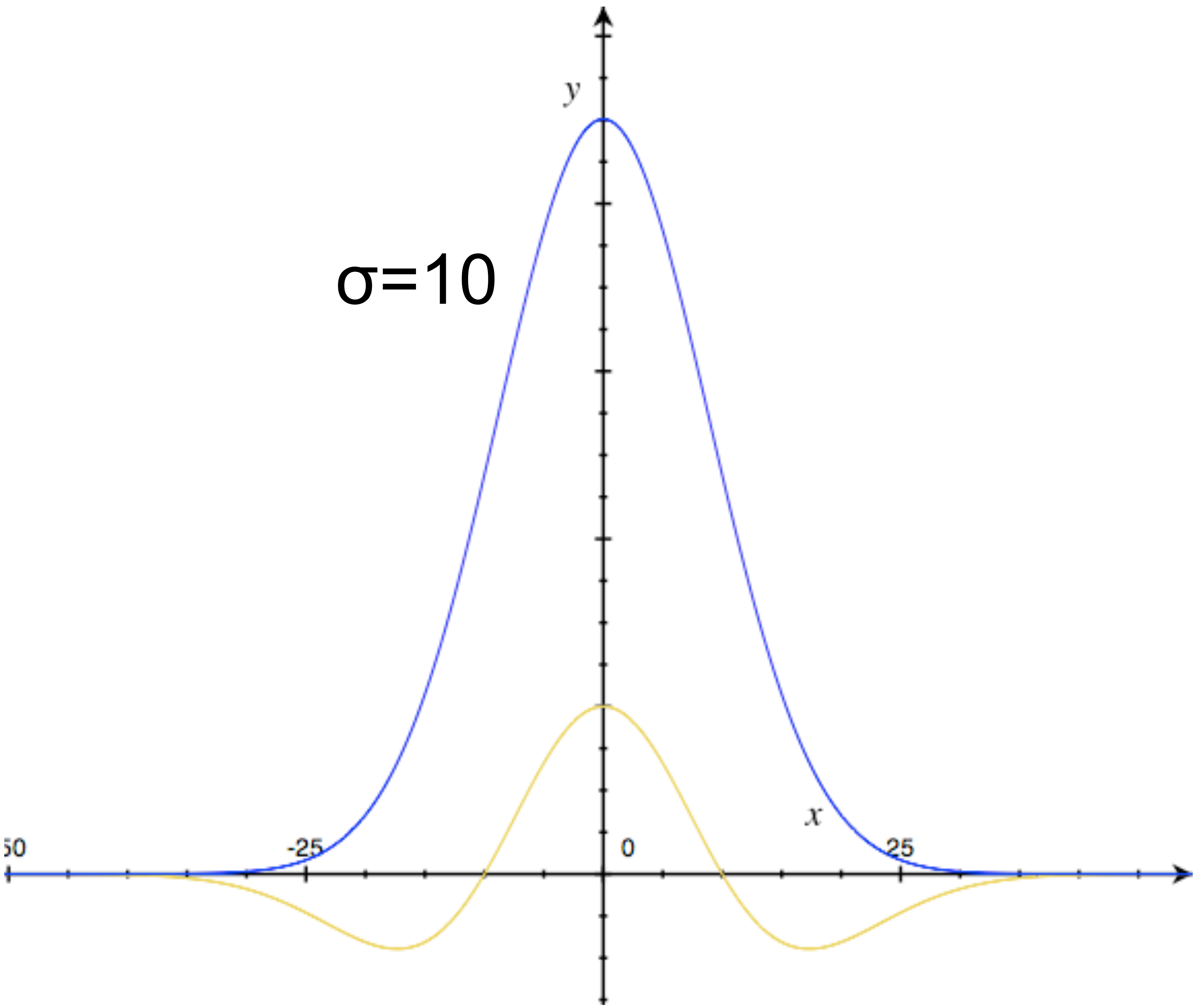
The Laplacian of Gaussian and arbitrary *z*-crossings approach applied to automated single particle reconstruction

David Woolford [a], Ben Hankamer [a], Geoffery Ericksson [b,*]

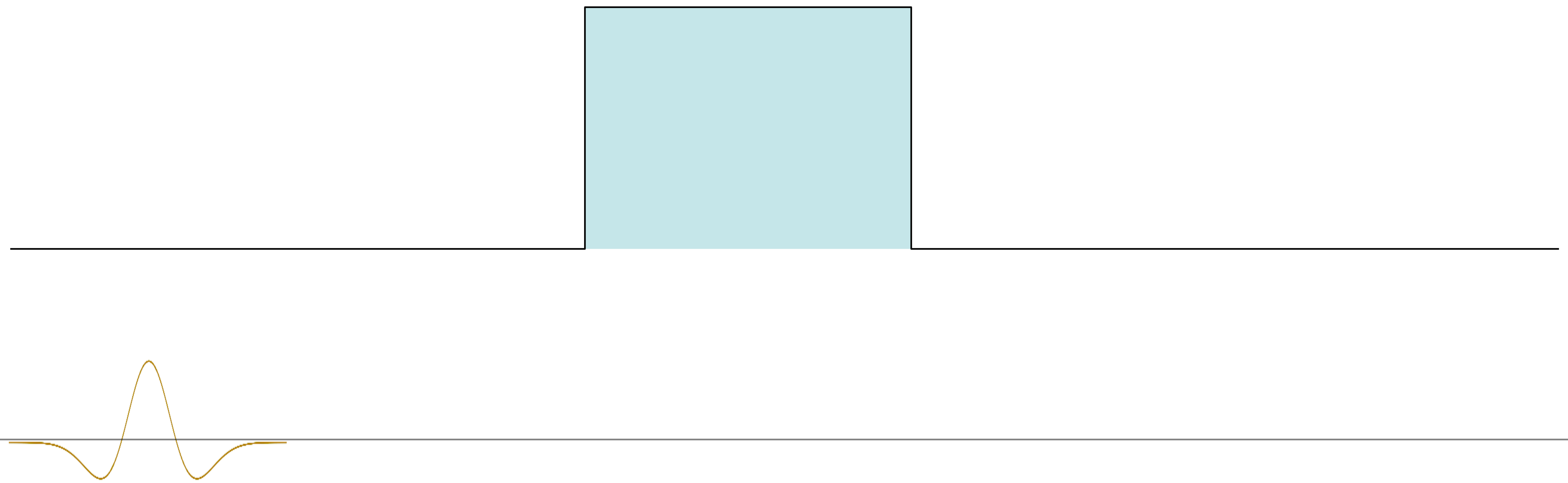
[a] *Institute for Molecular Bioscience, Brisbane, Qld 4072, Australia*
[b] *Queensland Brain Institute, University of Queensland, Brisbane, Qld 4072, Australia*

Received 21 November 2006; received in revised form 7 March 2007; accepted 7 March 2007

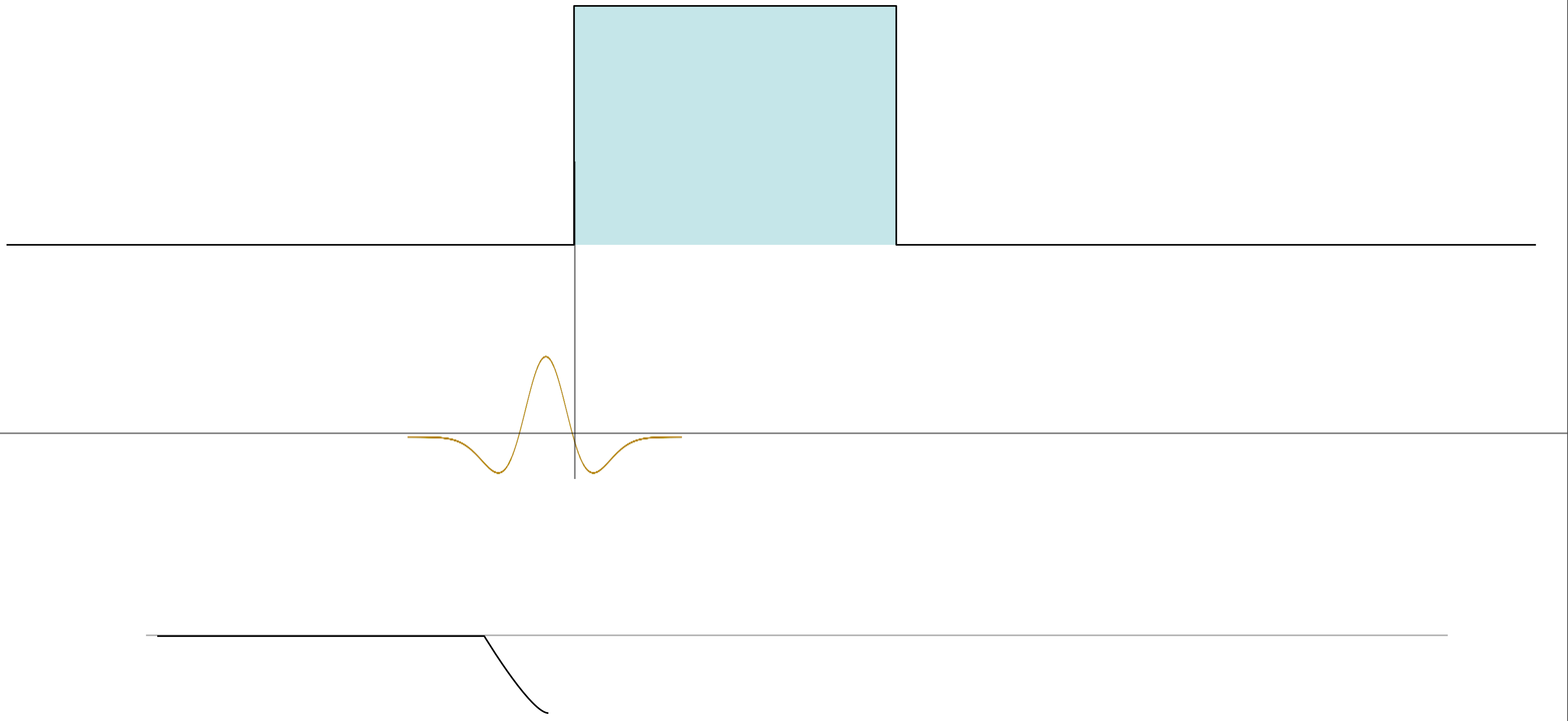
Gaussian Curves



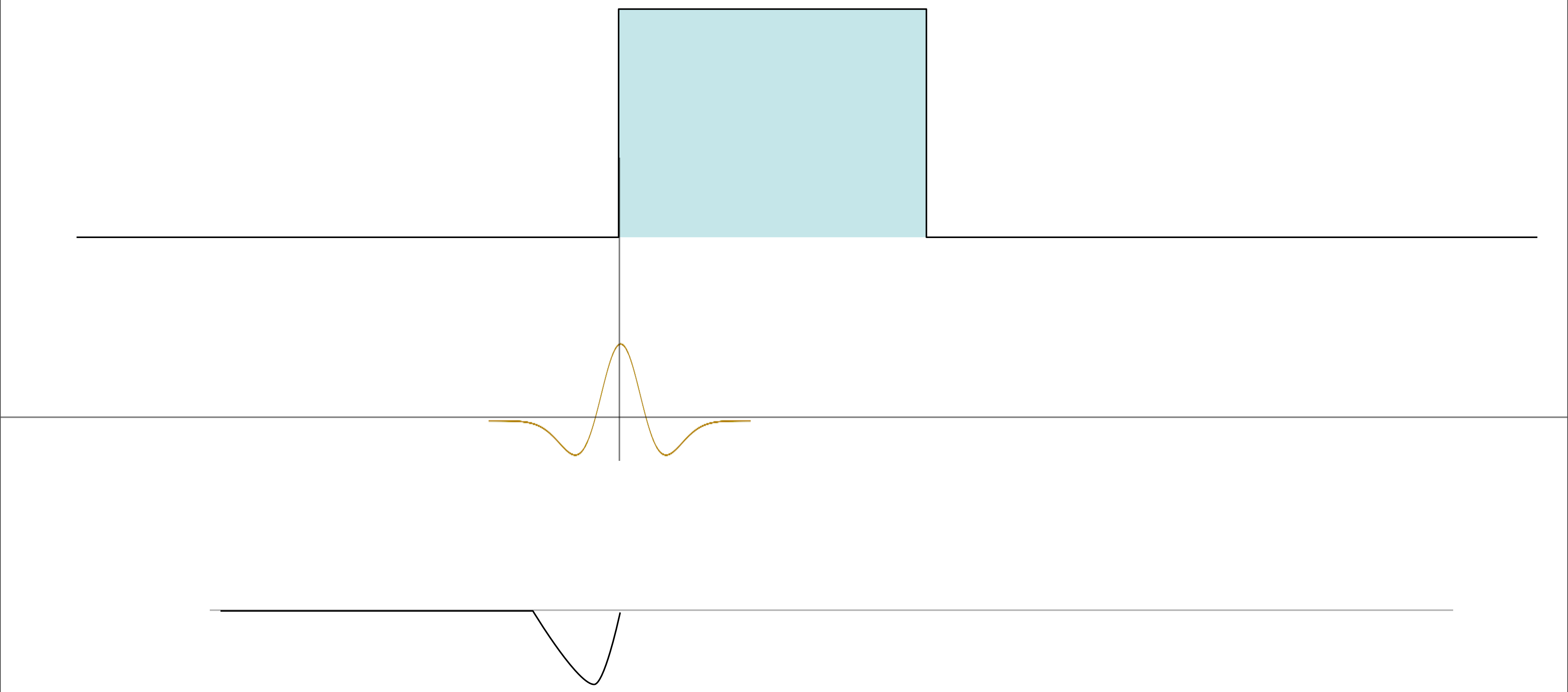
LoG Edge Finding



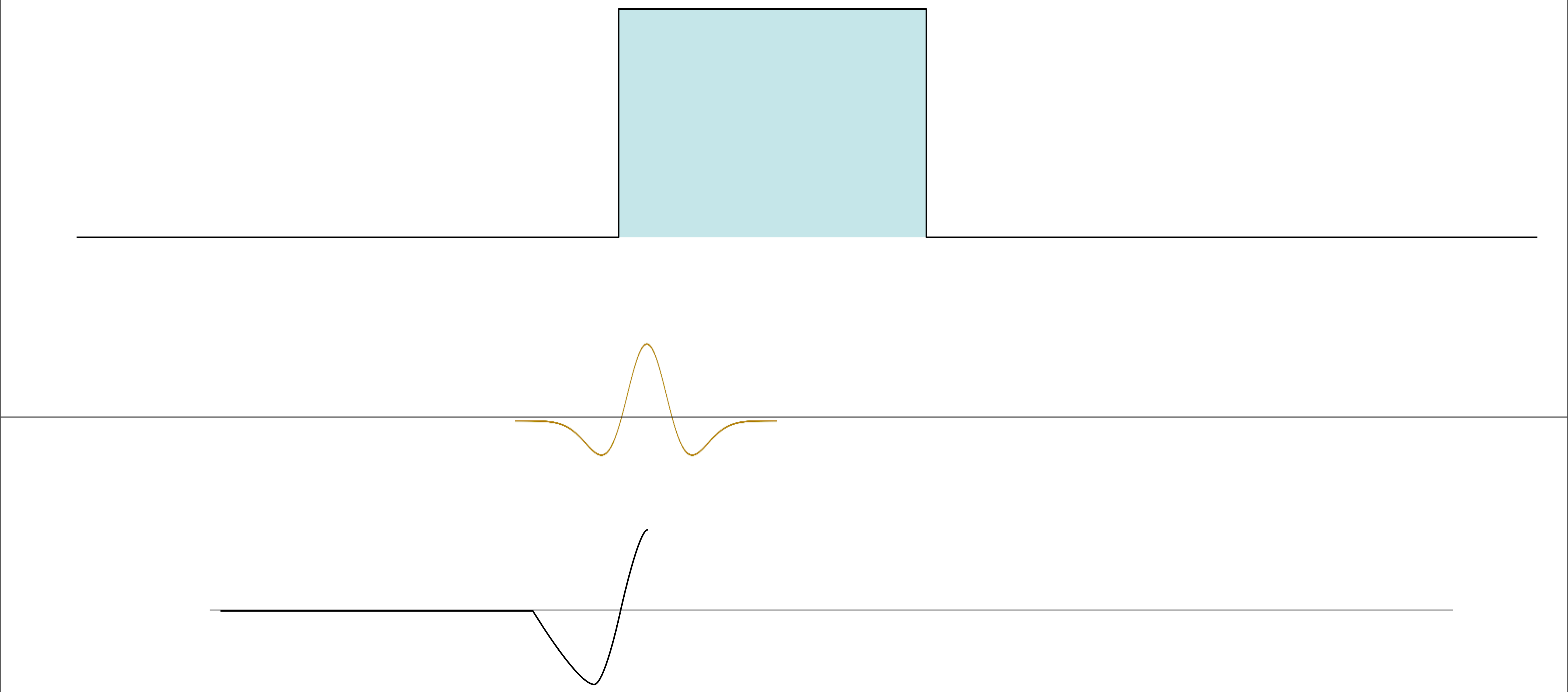
LoG Edge Finding



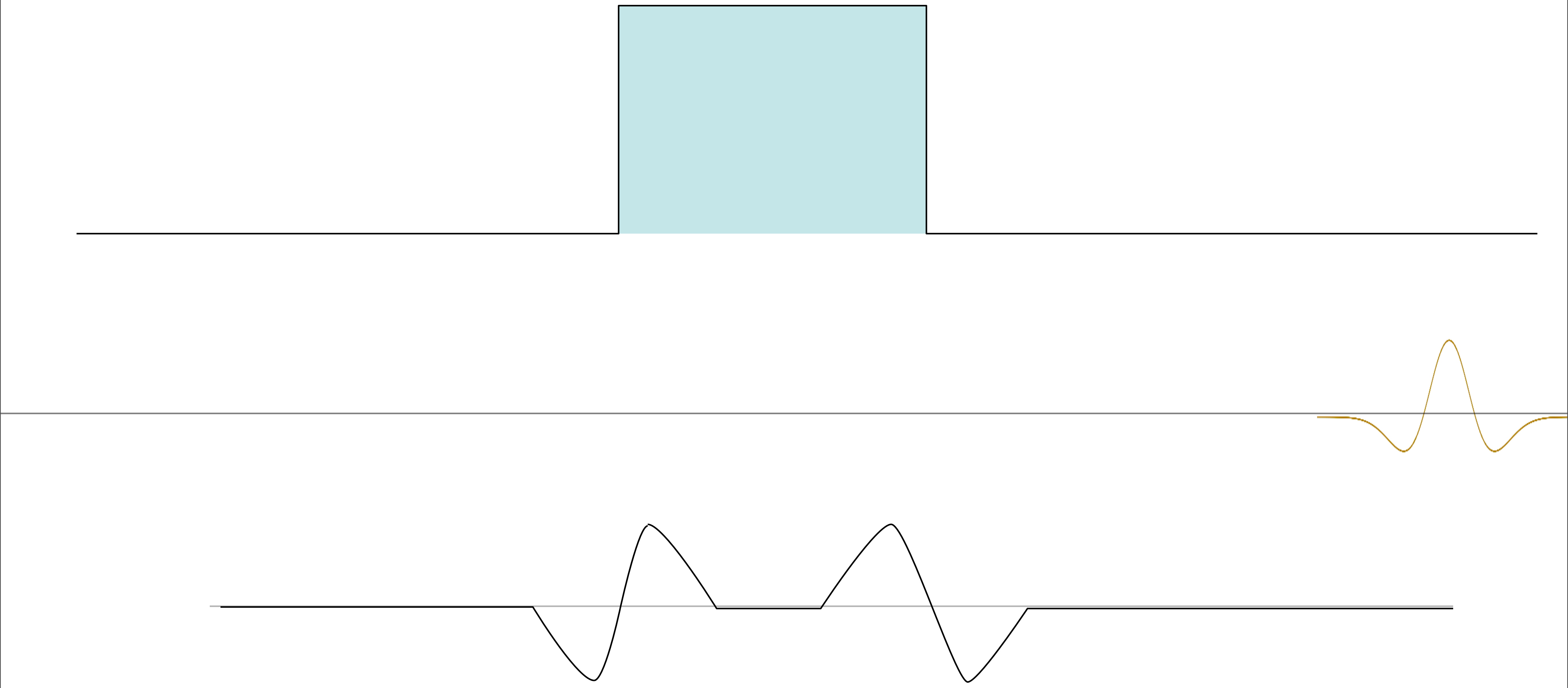
LoG Edge Finding



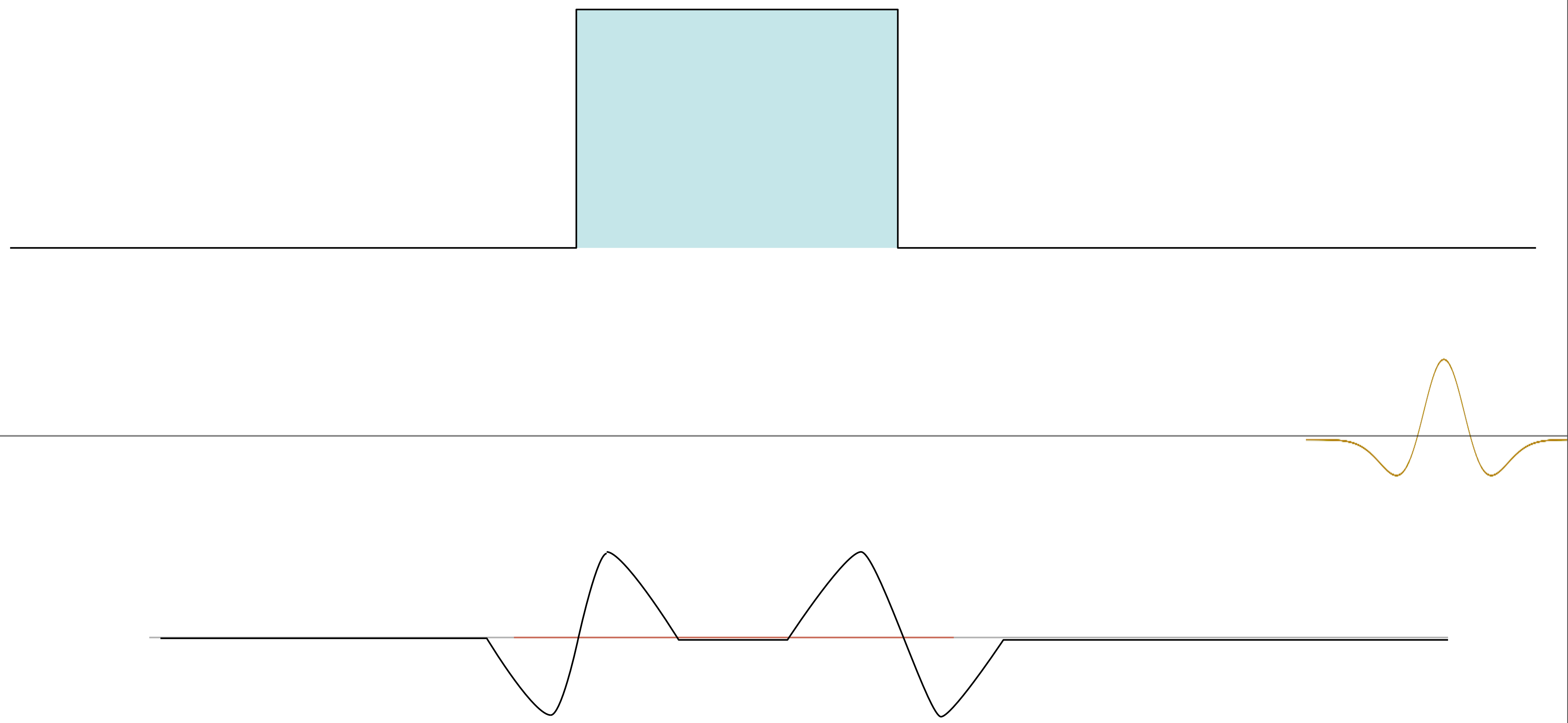
LoG Edge Finding



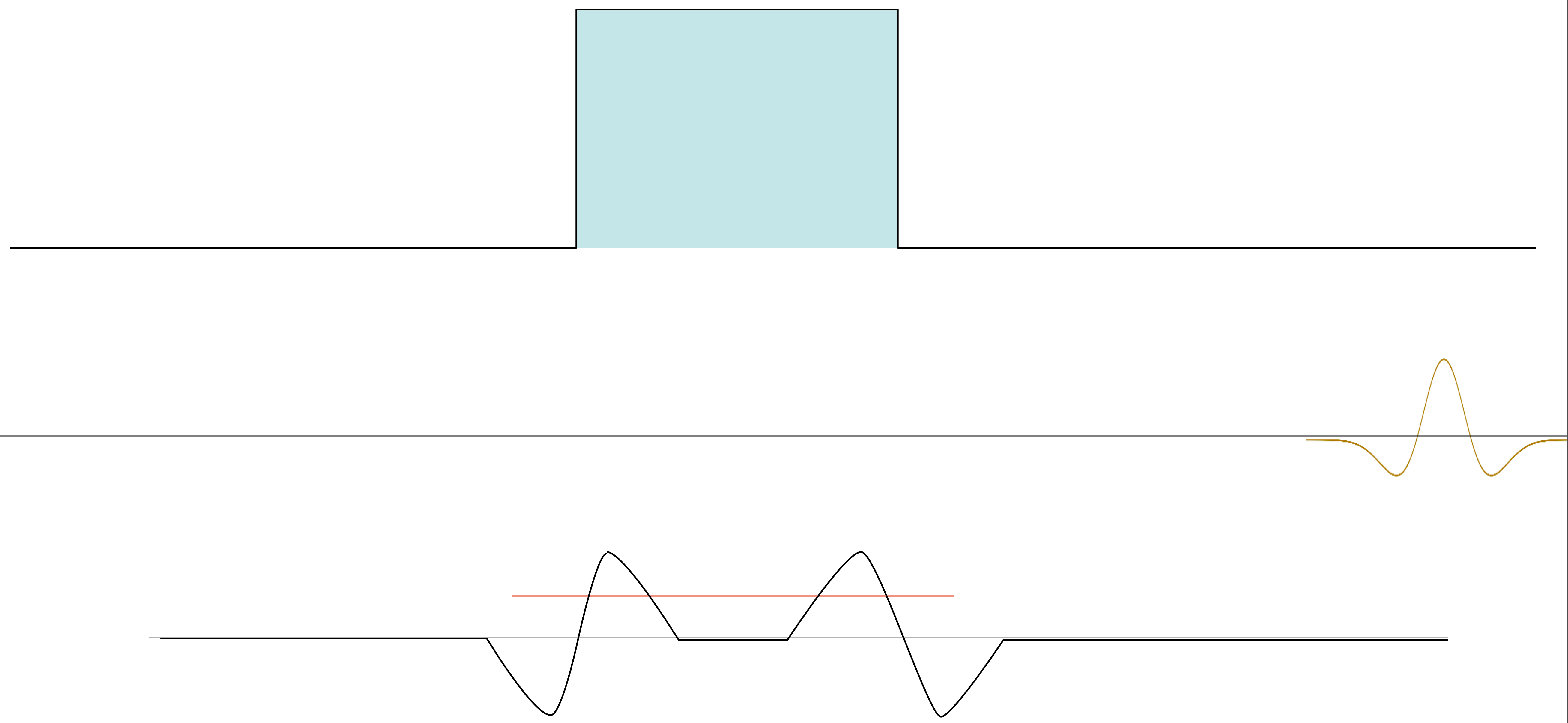
LoG Edge Finding



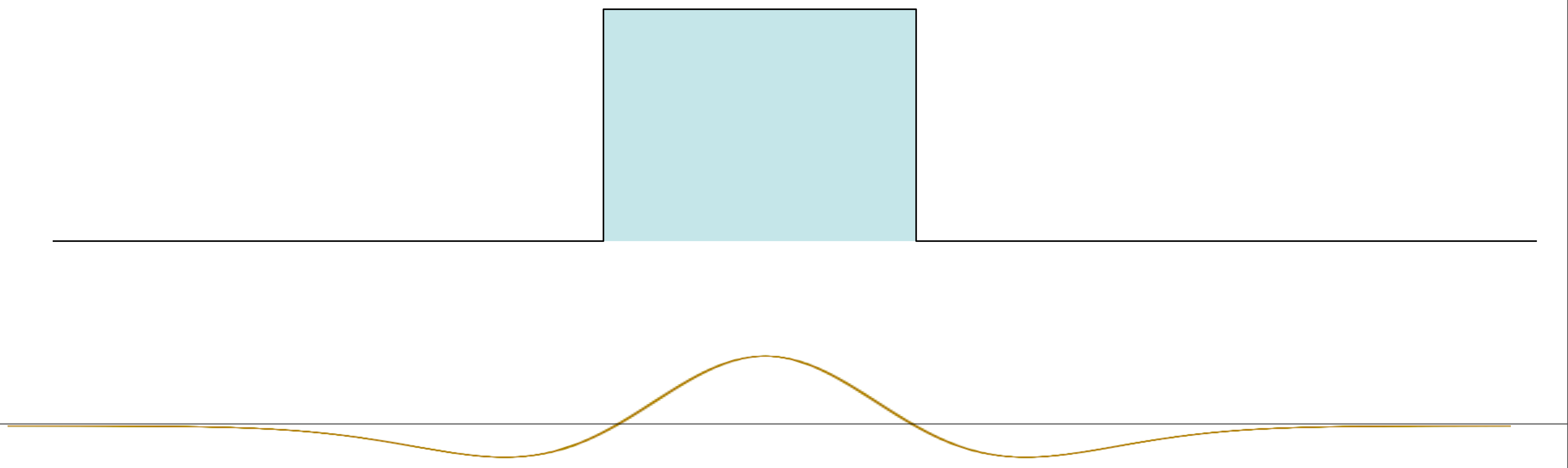
LoG Edge Finding



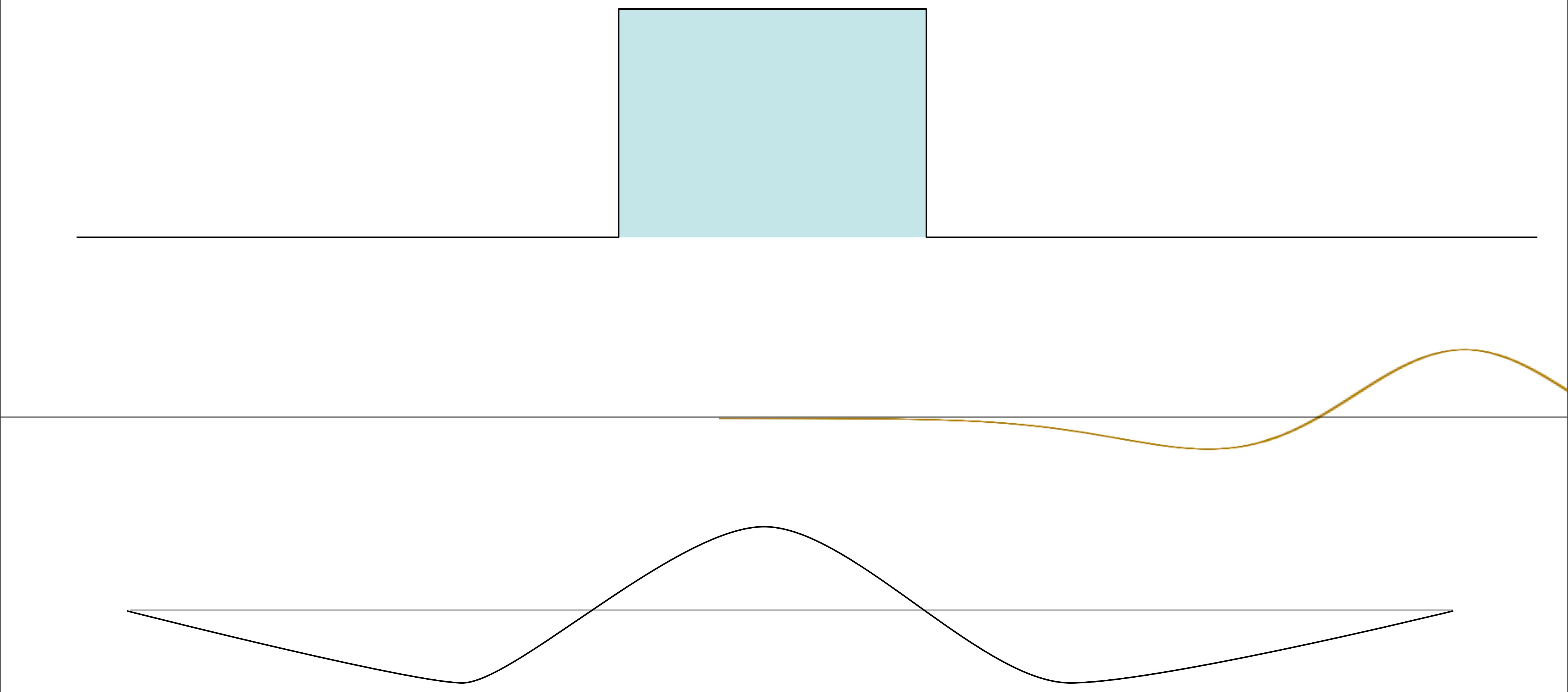
LoG Edge Finding



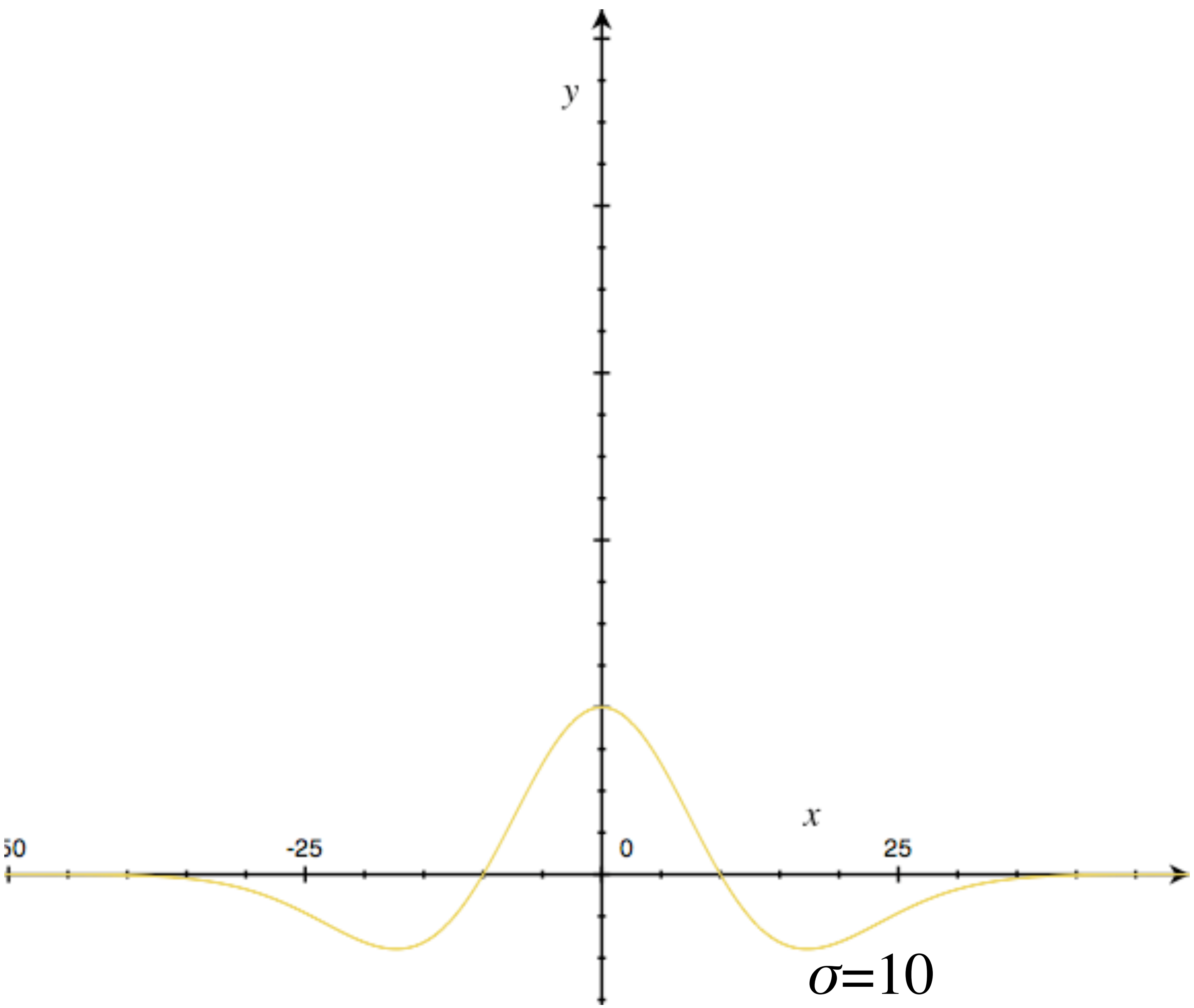
LoG Blob Finding



LoG Blob Finding

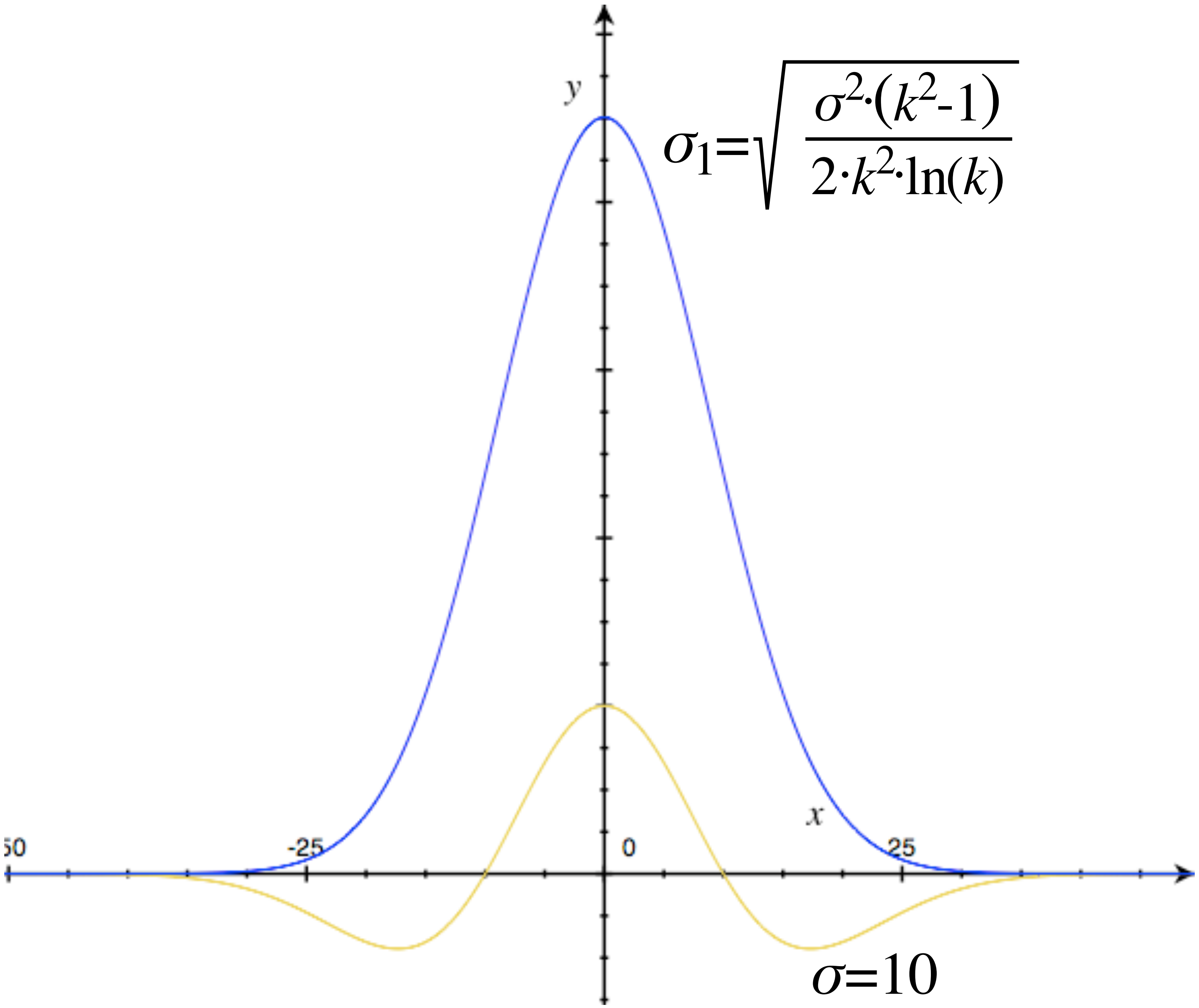


DoG, LoG, =?

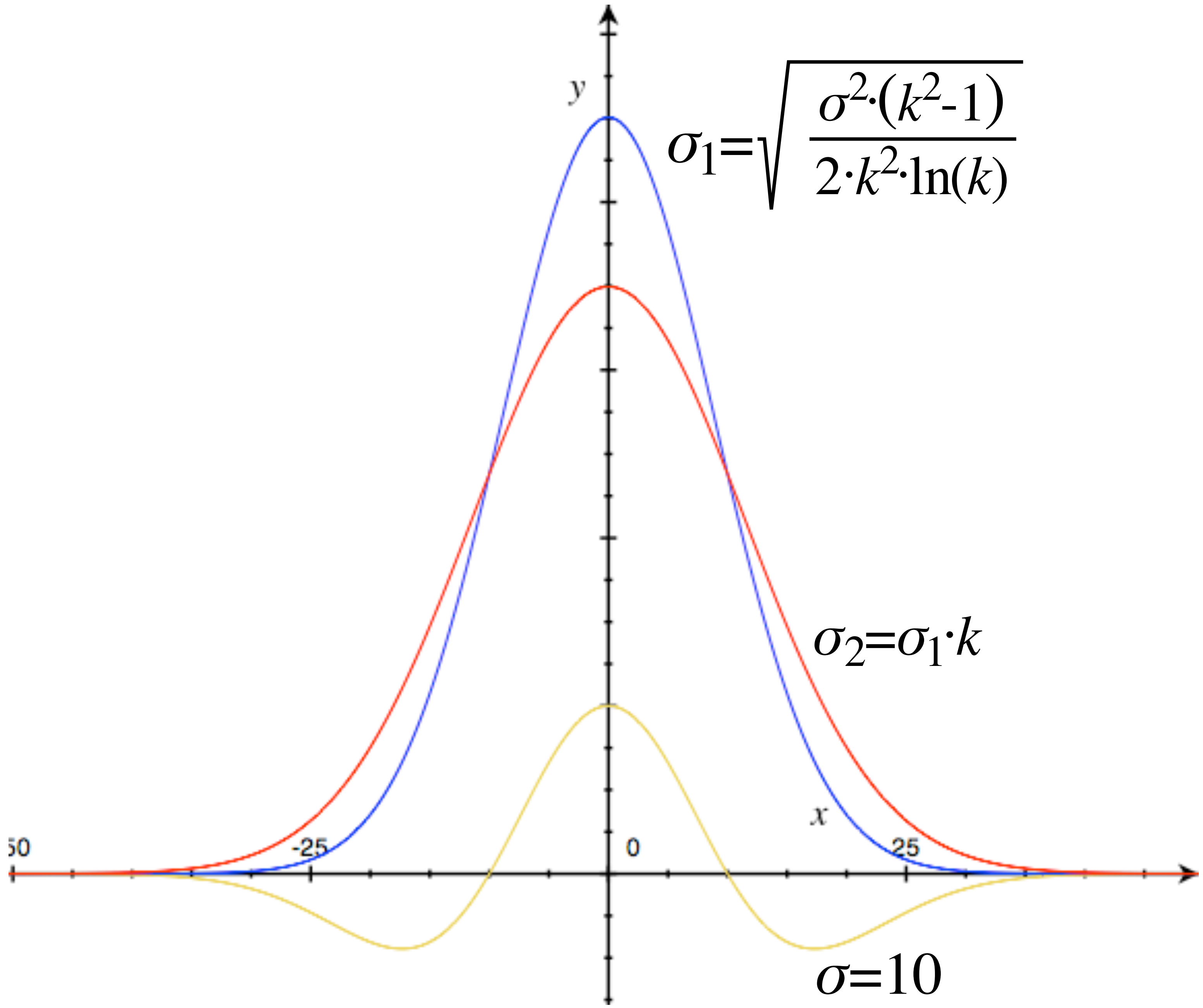


σ=10

DoG, LoG, =?

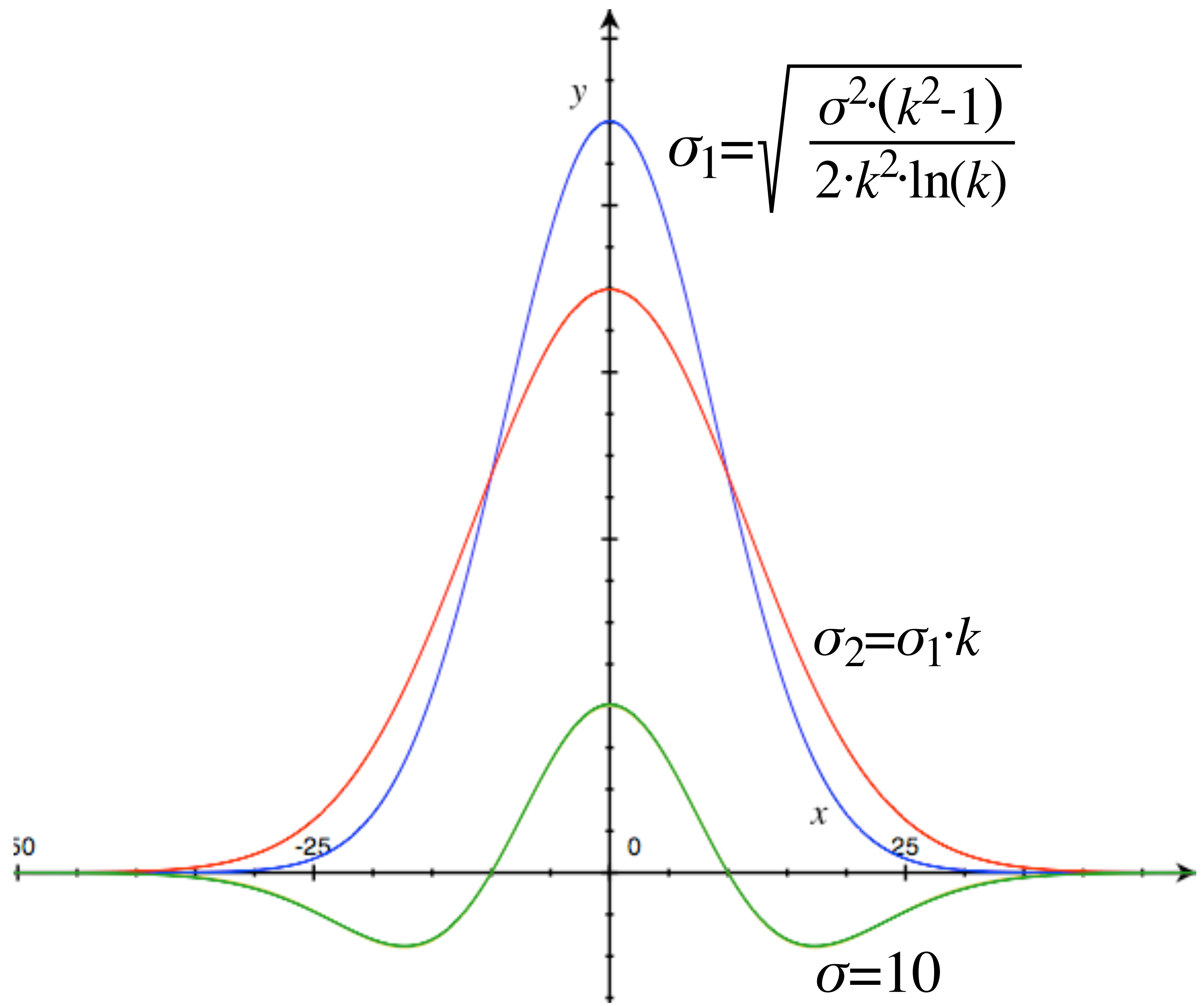

y
$\sigma_1=\sqrt{\frac{\sigma^2\cdot(k^2-1)}{2\cdot k^2\cdot\ln(k)}}$
x
-25
0
25
$\sigma=10$

DoG, LoG, =?



y
$\sigma_1=\sqrt{\frac{\sigma^2\cdot(k^2-1)}{2\cdot k^2\cdot \ln(k)}}$
$\sigma_2=\sigma_1\cdot k$
x
-25
0
25
$\sigma=10$

DoG, LoG, =?



y
$\sigma_1=\sqrt{\dfrac{\sigma^2\cdot(k^2\text{-}1)}{2\cdot k^2\cdot \ln(k)}}$
$\sigma_2=\sigma_1\cdot k$
x
-50
-25
0
25
$\sigma=10$

DoG≈LoG

DoG ≈ LoG

Search Range

$\sigma_1 = 10.0$

$\sigma_2 = 11.0$

$\sigma_3 = 12.1$

$\sigma_4 = 13.3$

$\sigma_5 = 14.6$

$\sigma_6 = 16.1$

$\sigma_7 = 17.7$

$\sigma_8 = 19.5$

$\sigma_9 = 21.4$

DoG ≈ LoG

Search Range

$\sigma_1 = 10.0$

$\sigma_2 = 11.0$

$\sigma_3 = 12.1$

$\sigma_4 = 13.3$

$\sigma_5 = 14.6$

$\sigma_6 = 16.1$

$\sigma_7 = 17.7$

$\sigma_8 = 19.5$

$\sigma_9 = 21.4$

The Shortcut:

$$\sigma_n = \sqrt{\sigma_1^2 + \sigma_2^2 + \sigma_3^2 + \ldots + \sigma_{n-1}^2}$$

$$\sigma_n = \sigma_{n-1}\sqrt{k^2 - 1}$$

DoG≈LoG

Search Range

$\sigma_1 = 10.0$
$\sigma_2 = 11.0$
$\sigma_3 = 12.1$
$\sigma_4 = 13.3$
$\sigma_5 = 14.6$
$\sigma_6 = 16.1$
$\sigma_7 = 17.7$
$\sigma_8 = 19.5$
$\sigma_9 = 21.4$

The Shortcut:

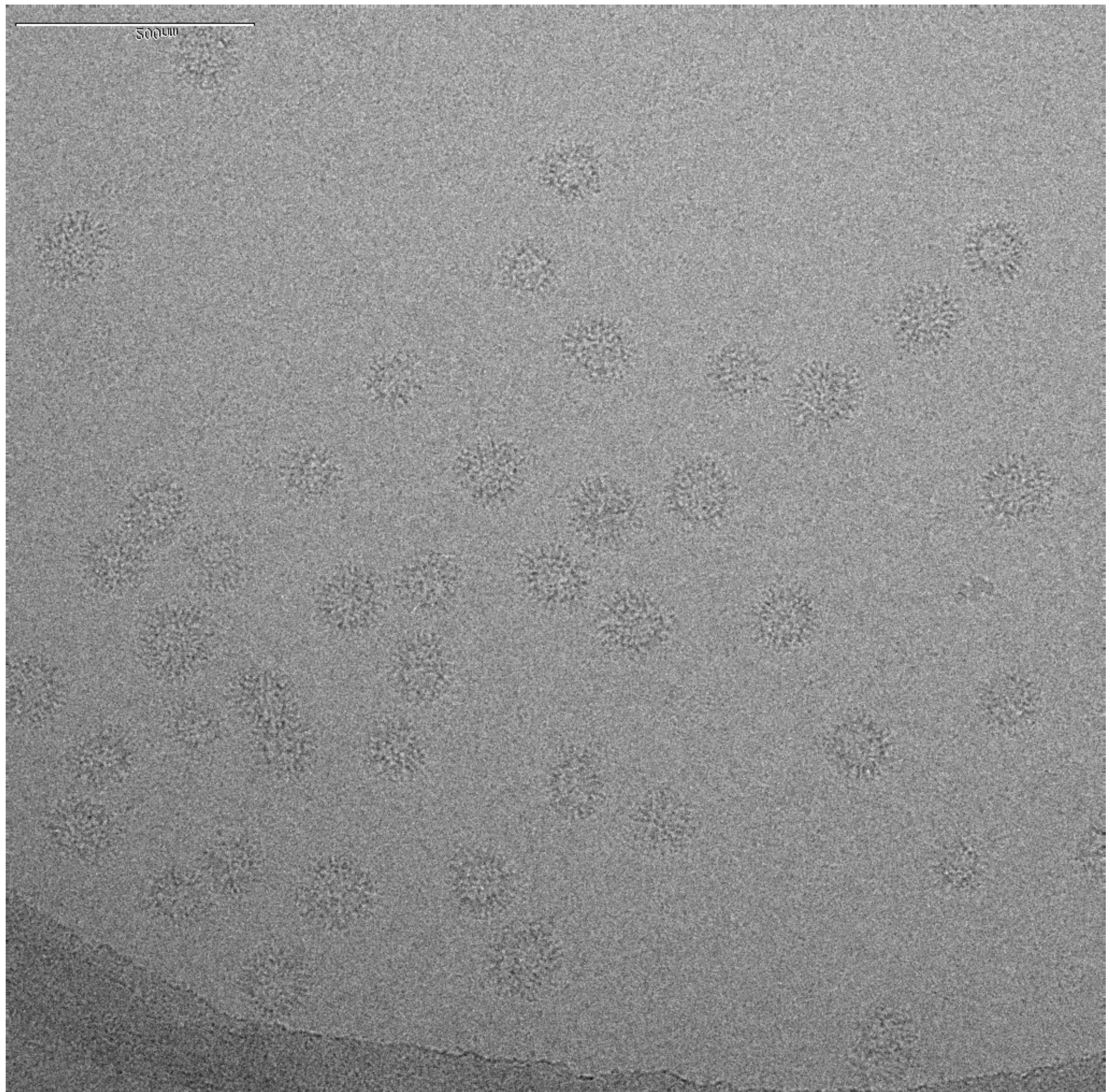
$$\sigma_n = \sqrt{\sigma_1^2 + \sigma_2^2 + \sigma_3^2 + \ldots + \sigma_{n-1}^2}$$

$$\sigma_n = \sigma_{n-1}\sqrt{k^2 - 1}$$

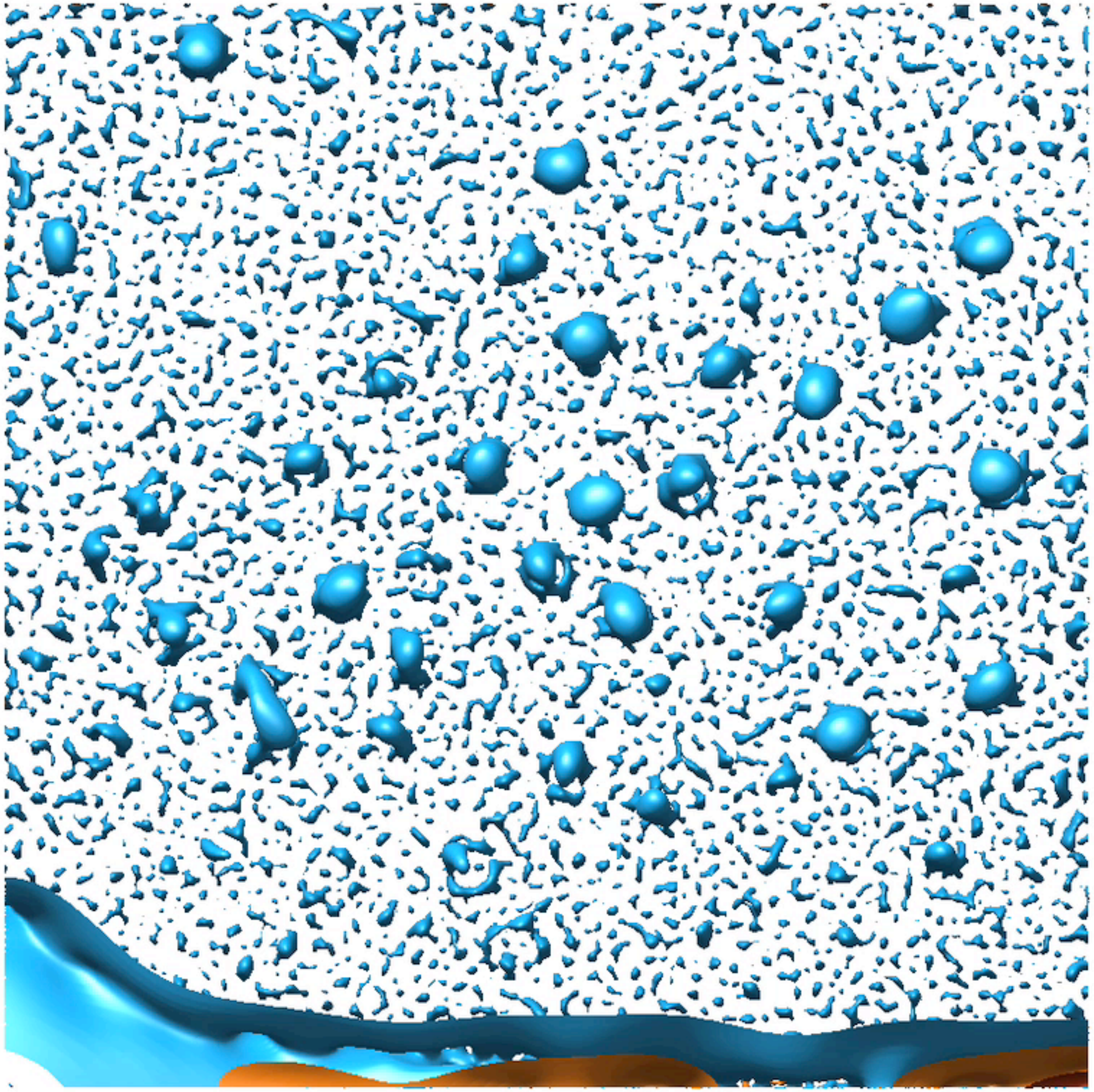
Cascaded Blurs

$\sigma_1 = 10.0$
$\sigma_2 = 4.6$
$\sigma_3 = 5.0$
$\sigma_4 = 5.5$
$\sigma_5 = 6.1$
$\sigma_6 = 6.7$
$\sigma_7 = 7.4$
$\sigma_8 = 8.1$
$\sigma_9 = 8.9$

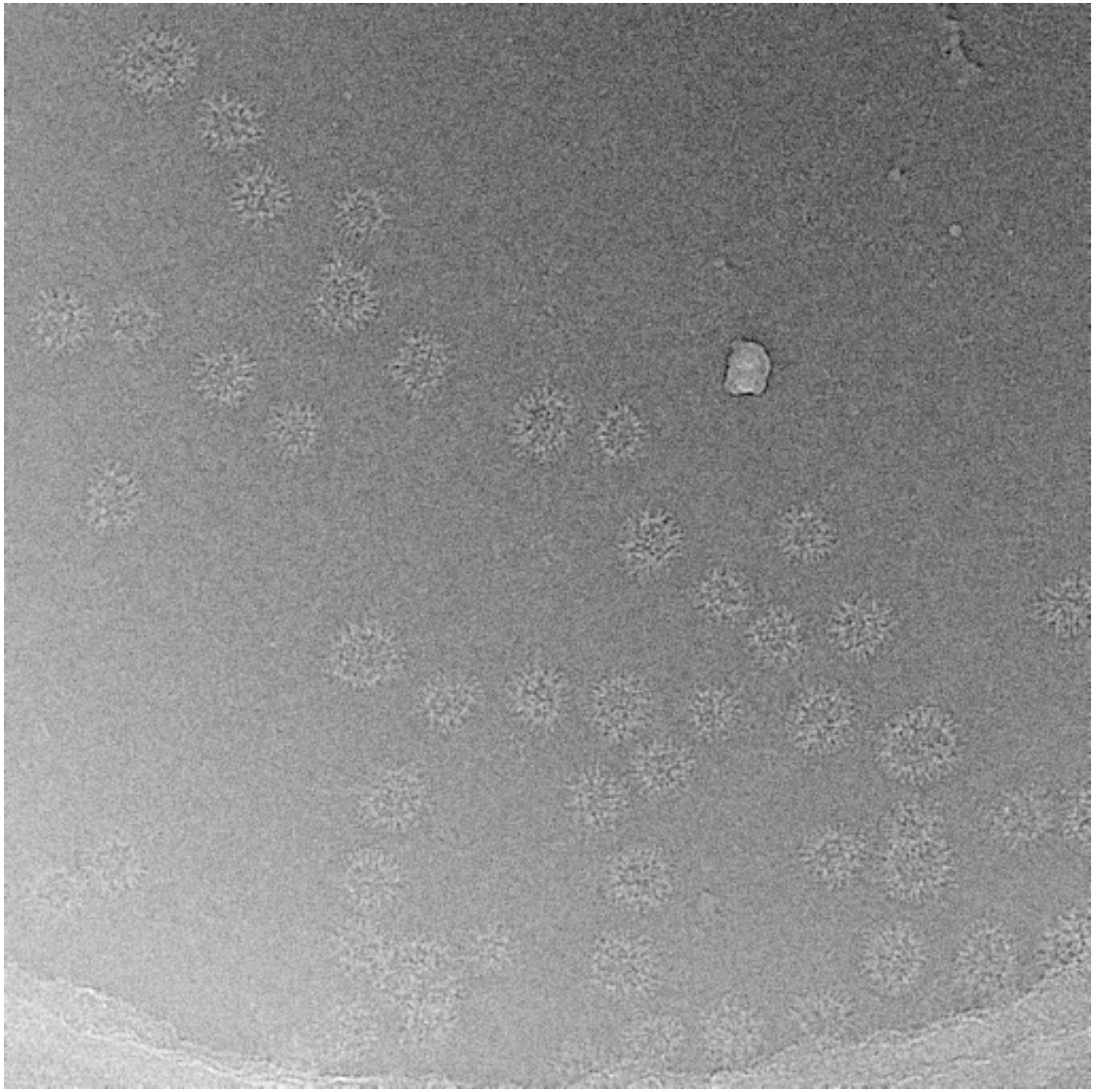
The DoG Scale Space



The DoG Scale Space

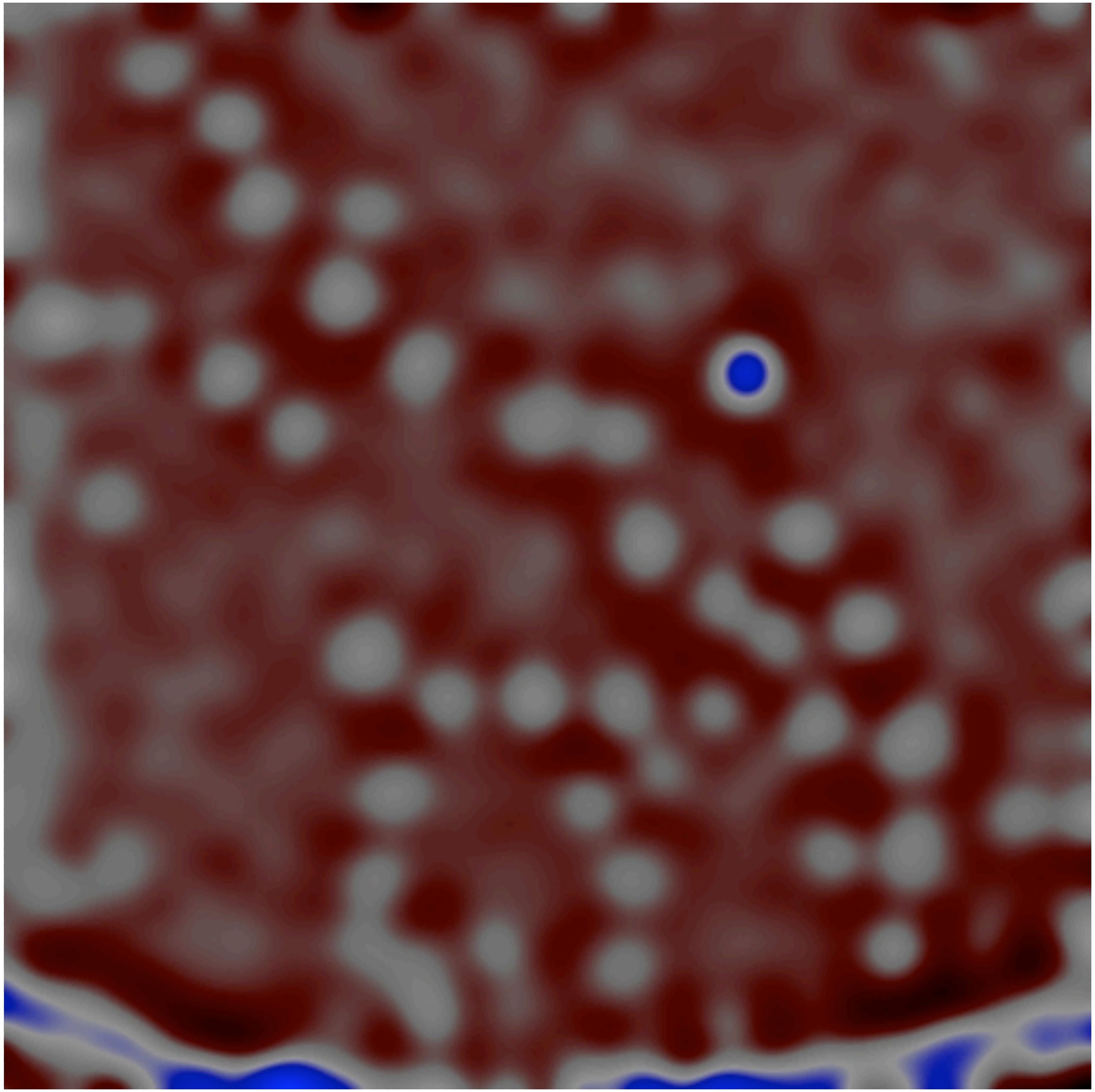


DoG Picker Examples



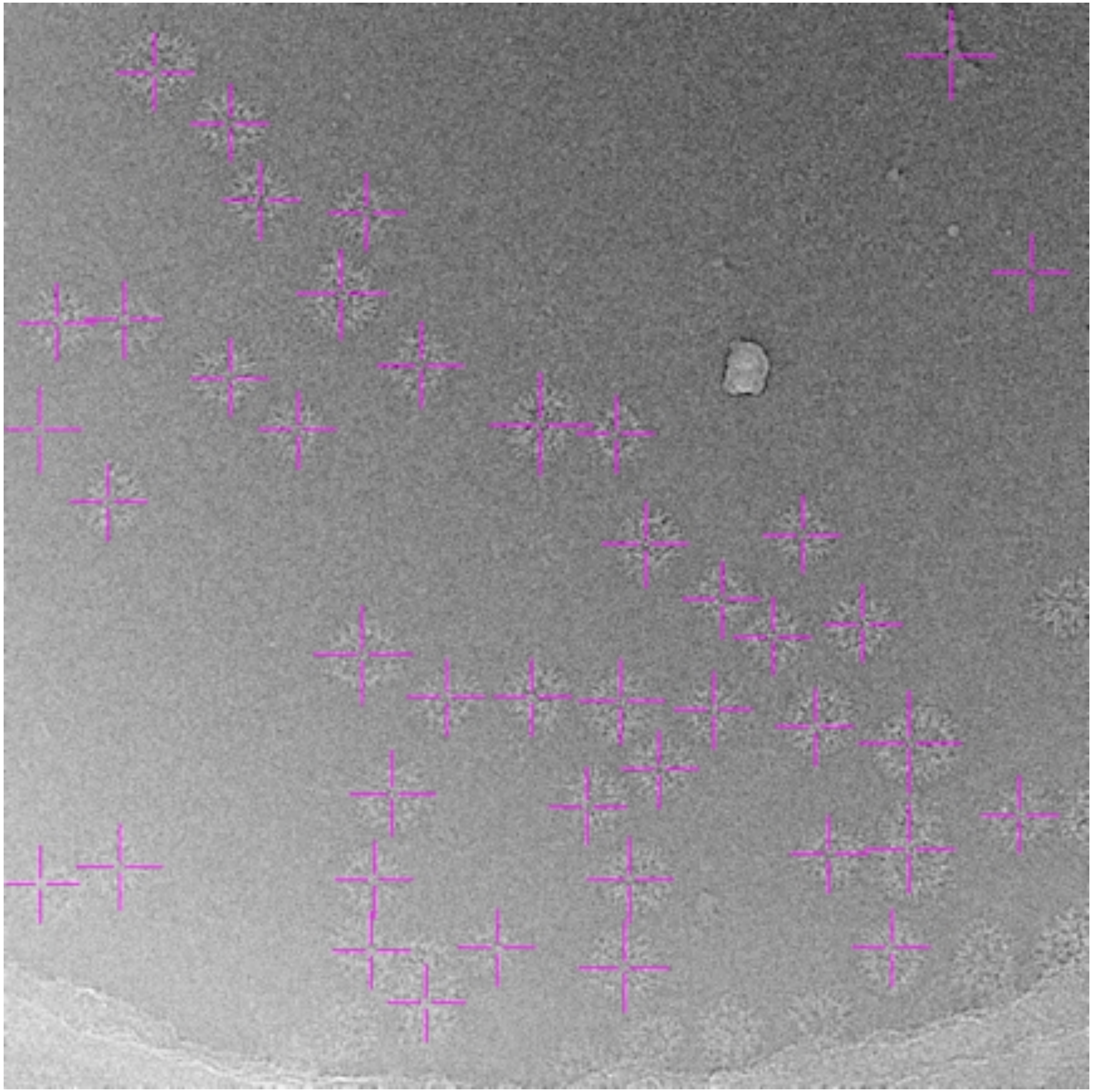
Virus Like Particles

DoG Picker Examples



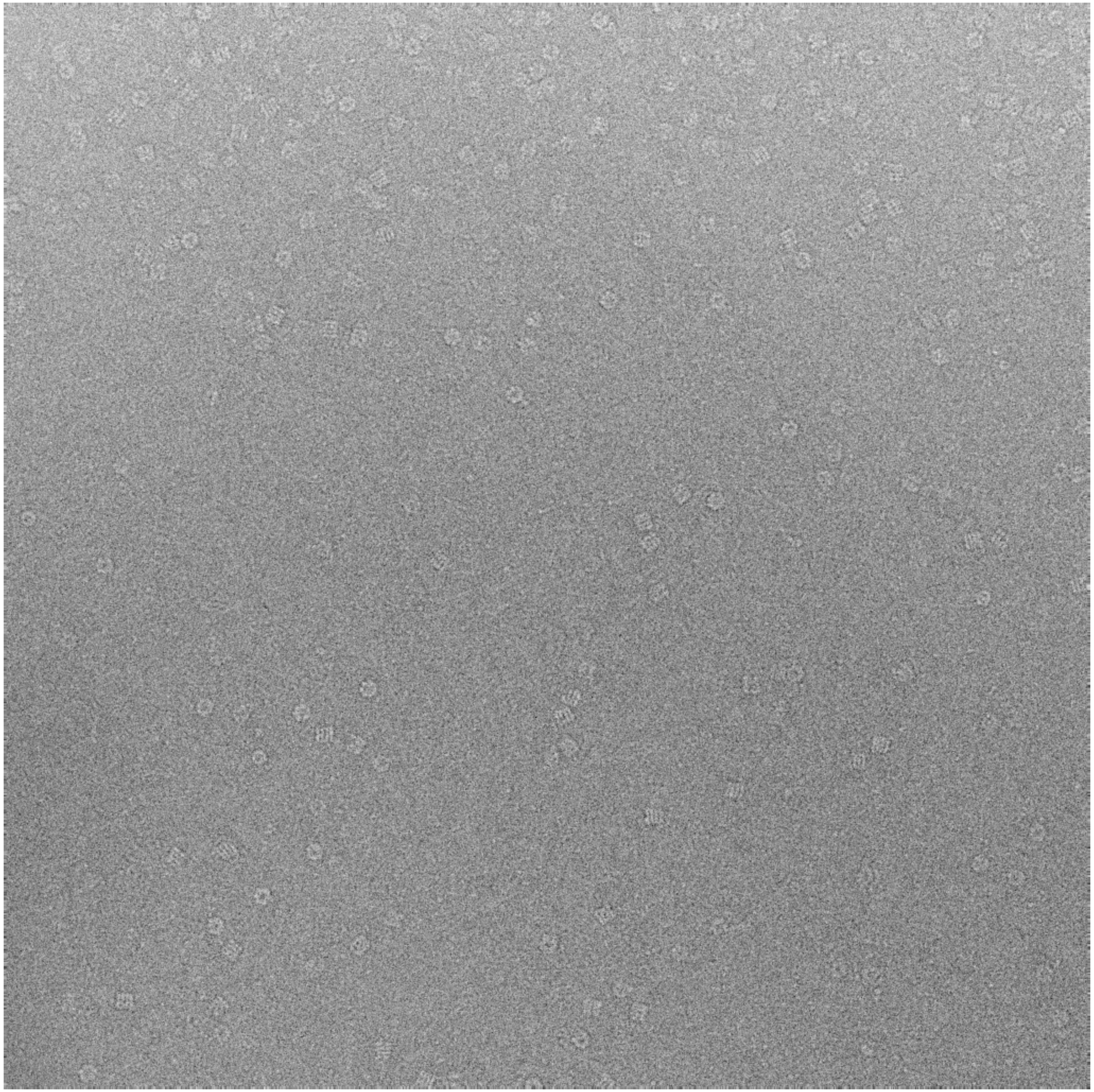
Virus Like Particles

DoG Picker Examples



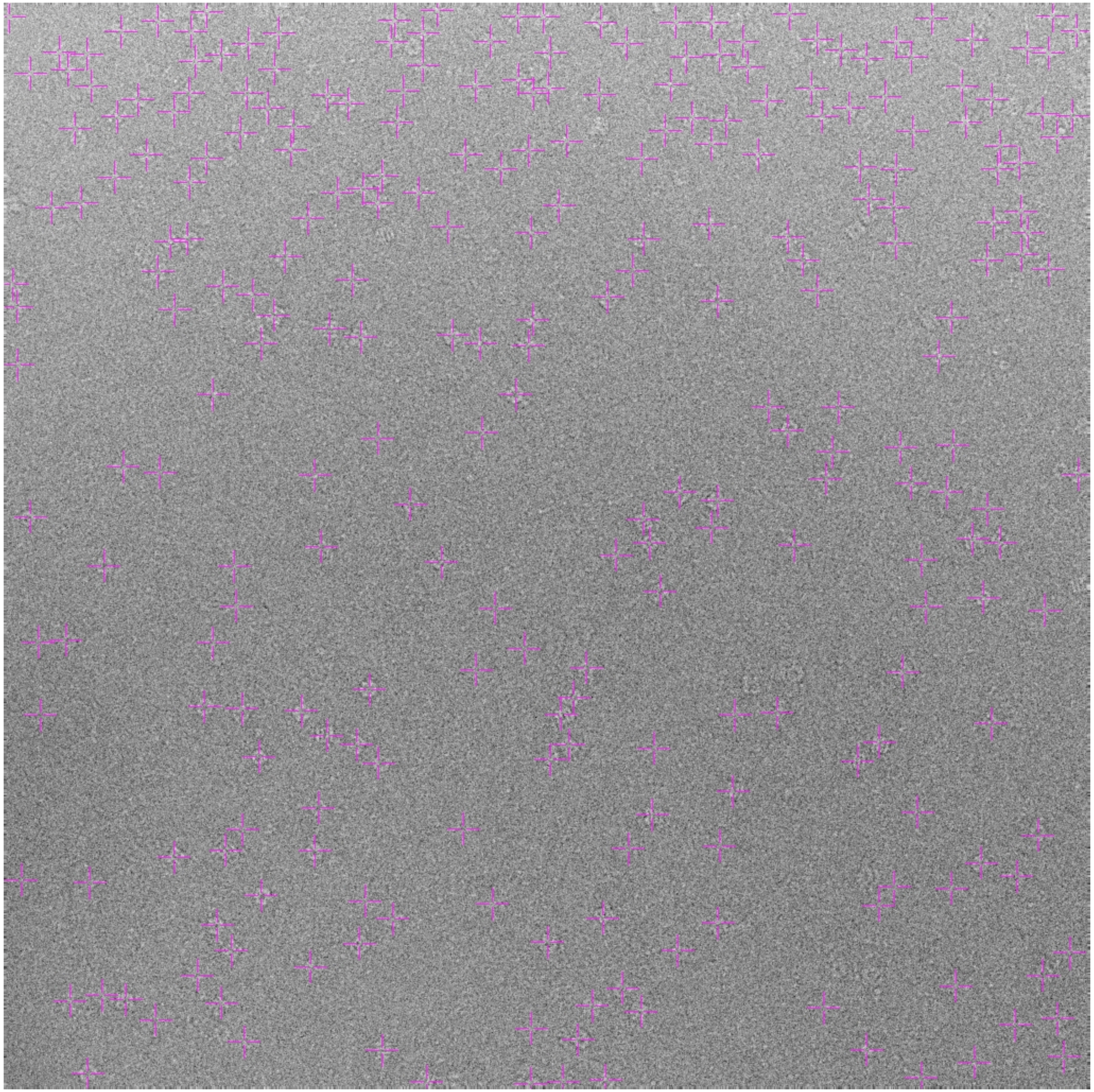
Virus Like Particles

DoG Picker Examples



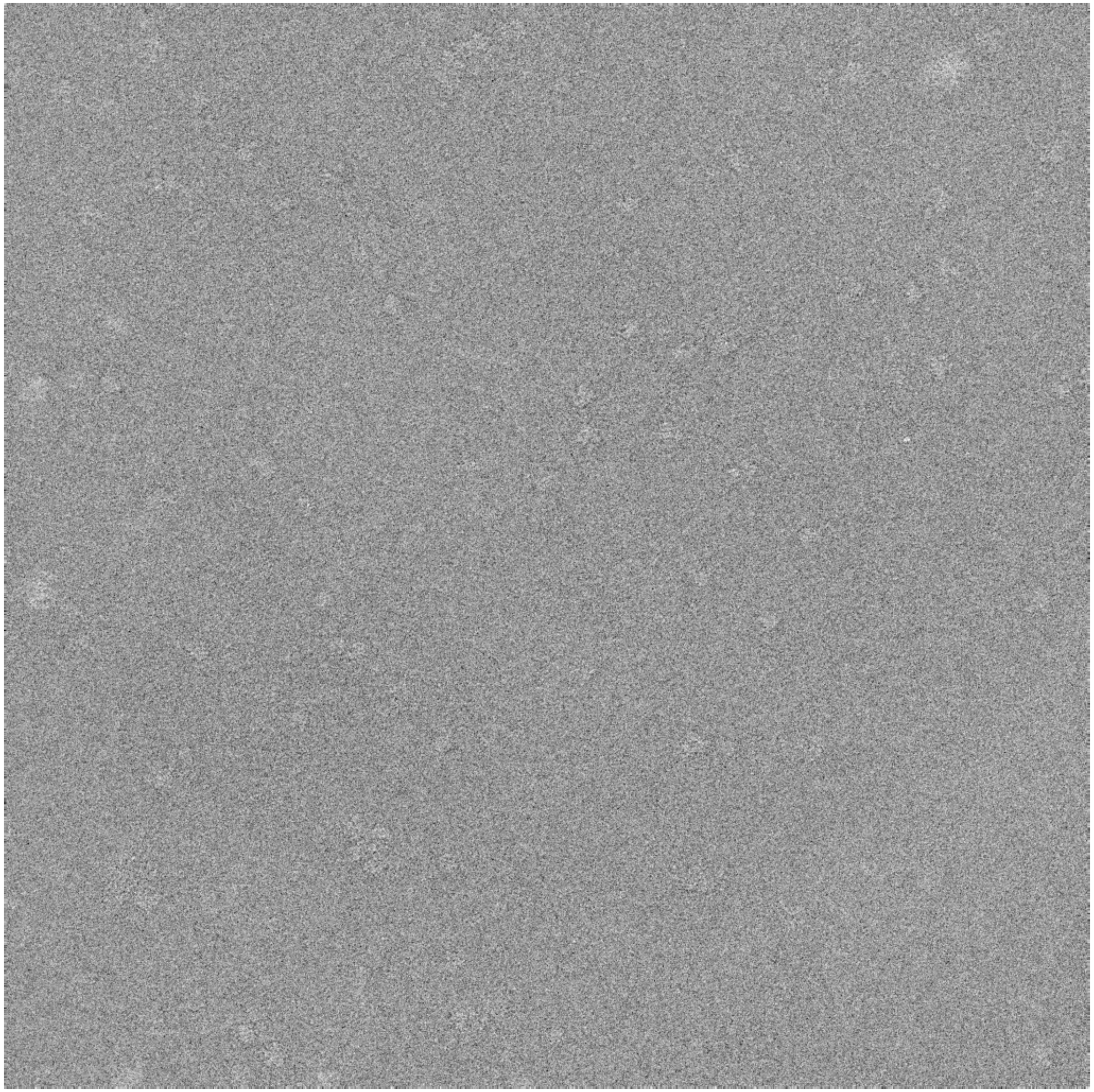
GroEL

DoG Picker Examples



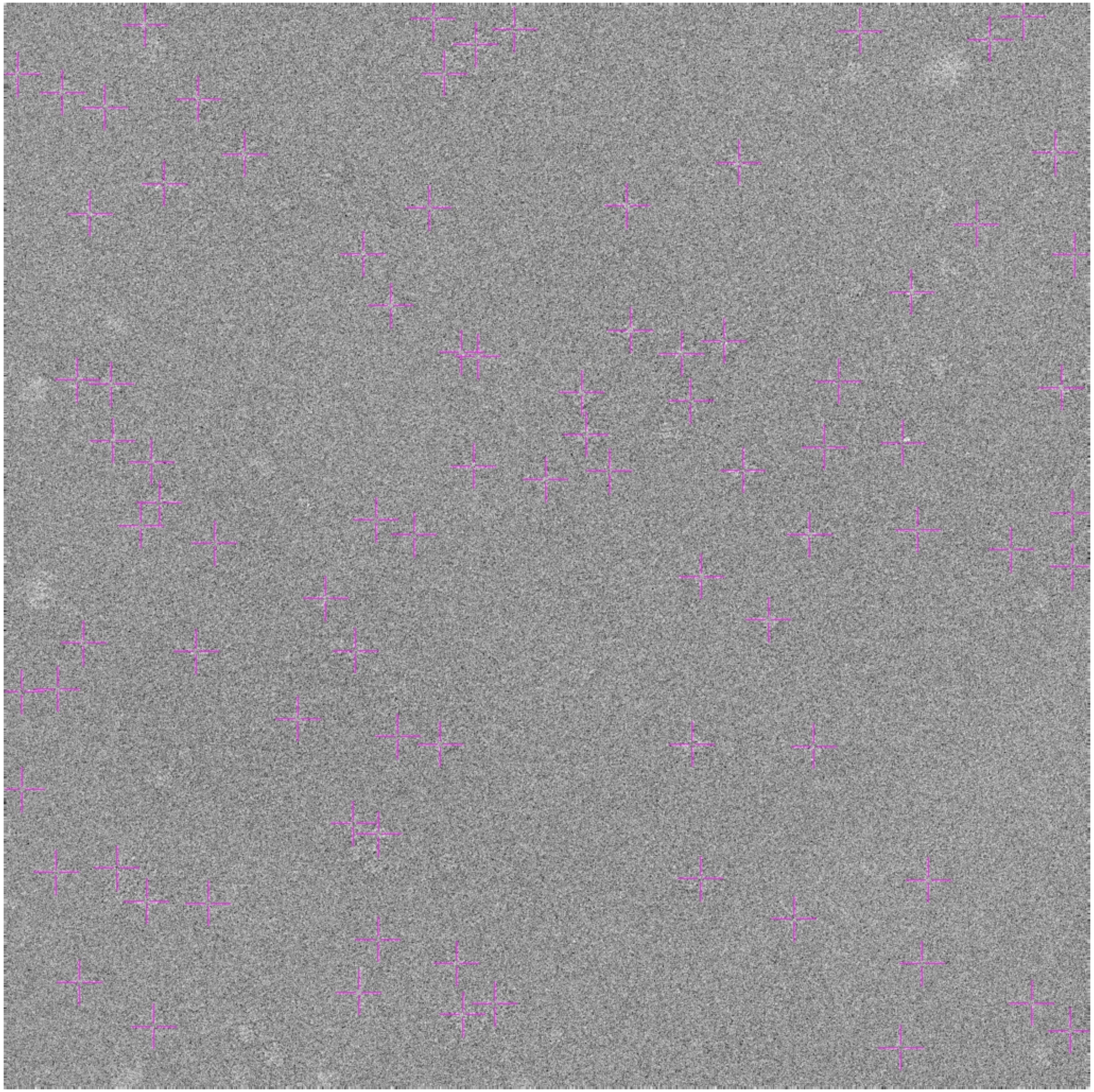
GroEL

DoG Picker Examples



Ribosomes

DoG Picker Examples



Ribosomes


Thank You