# Denoising with filters

D.A. Forsyth,

University of Illinois at Urbana Champaign

Some slides adapted from
Svetlana Lazebnik, who adapted from Alyosha Efros, Derek Hoiem

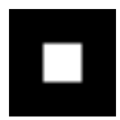
# A crucial property of images

- Pixels are like their neighbors
  (mostly, for most pixels)

- Imagine you wish to denoise an image.  You could do so by averaging neighbors (a filter!).

# Smoothing with box filter revisited

e?



Source: D. Forsyth

# A crucial property of images

- Pixels are like their neighbors
  (mostly, for most pixels)

- Imagine you wish to denoise an image.  You could do so by averaging neighbors (a filter!). Weighting the neighbors so nearby neighbors get heavier weights is a good move.

# Smoothing

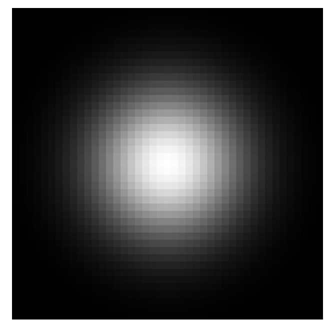
- To eliminate edge effects, weight contribution of neighborhood pixels according to their closeness to the center

“proportional to”
(renormalize values to sum to 1)

$$G(x, y) \propto \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

standard deviation
(determines size of “blob”)

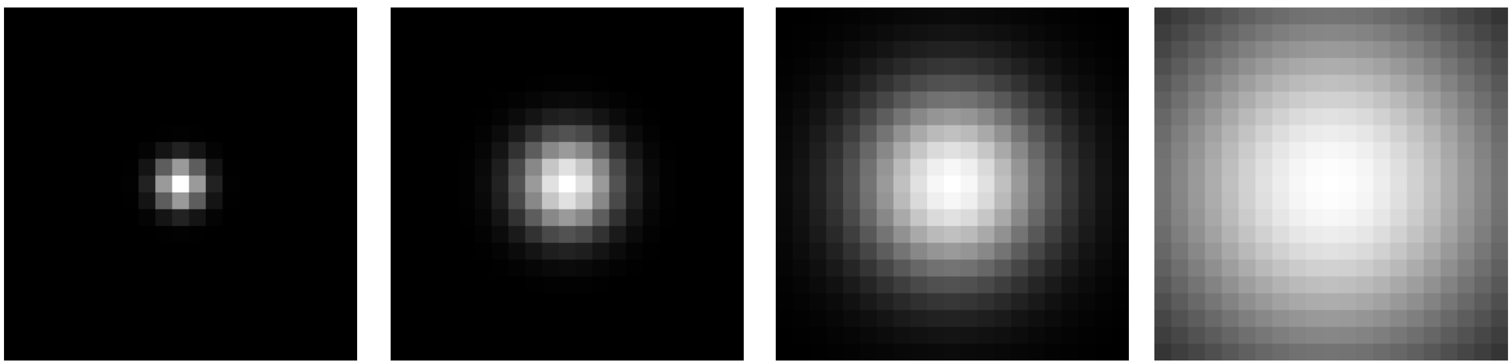
Gaussian filter



Adapted from D. Fouhey and J. Johnson

# Gaussian filters

$\sigma = 1$ $\sigma = 2$ $\sigma = 4$ $\sigma = 8$



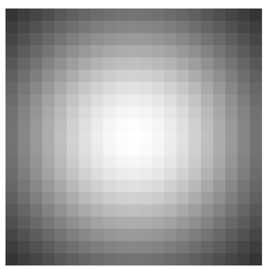
Filter size: $21 \times 21$

Adapted from D. Fouhey and J. Johnson

# Choosing filter size

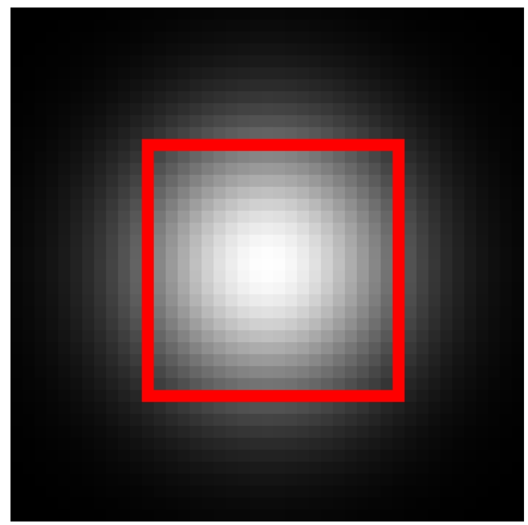
- Rule of thumb: set filter width to about $6\sigma$ (captures 99.7% of the energy)

$\sigma = 8$
$\text{Width} = 21$



Too small!

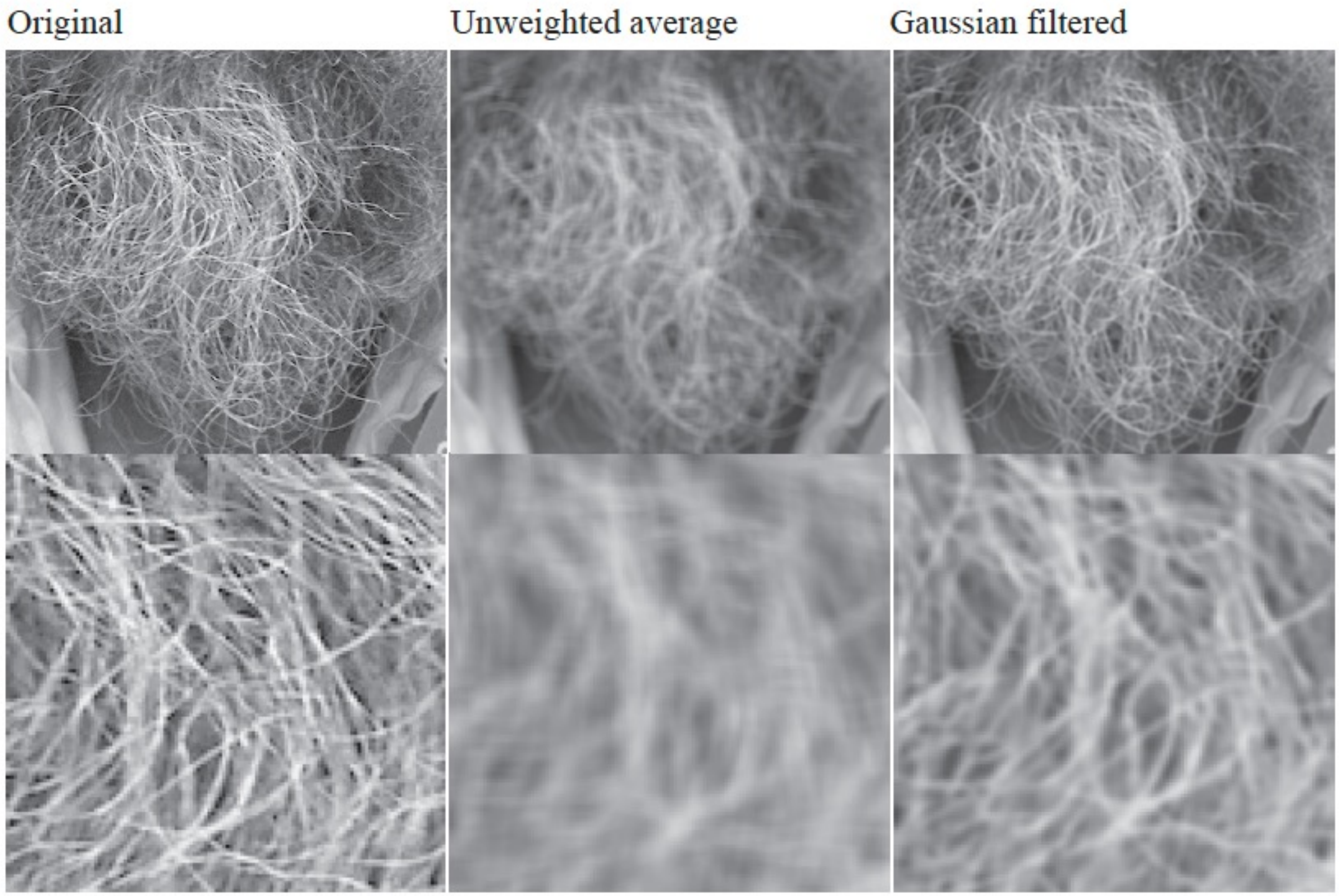
$\sigma = 8$
$\text{Width} = 43$



A bit small (might be OK)

Adapted from D. Fouhey and J. Johnson

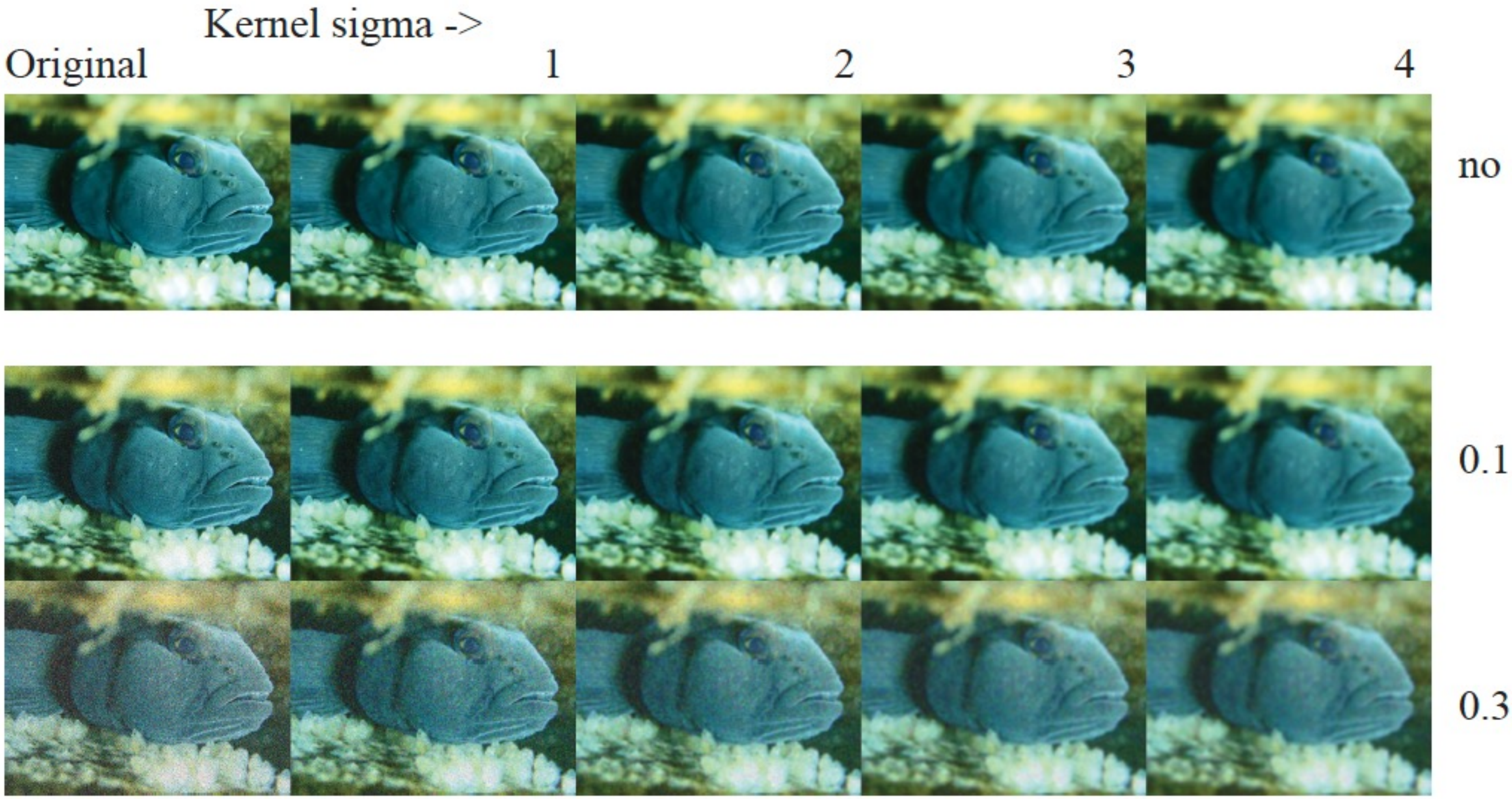
# Gaussian vs. box filtering



# Gaussian noise

The simplest model of image noise is the *additive stationary Gaussian noise* (or *Gaussian noise*) model, where each pixel has added to it a value chosen independently from the same normal (Gaussian – same Gauss, different sense) probability distribution. This distribution almost always has zero mean. The standard deviation is a parameter of the model. Figure 4.6 shows some examples of additive stationary Gaussian noise.

# Gaussian smoothing of Gaussian noise



# Smoothing by how much?

The choice of $\sigma$ (or scale) for the Gaussian follows from the following considerations:

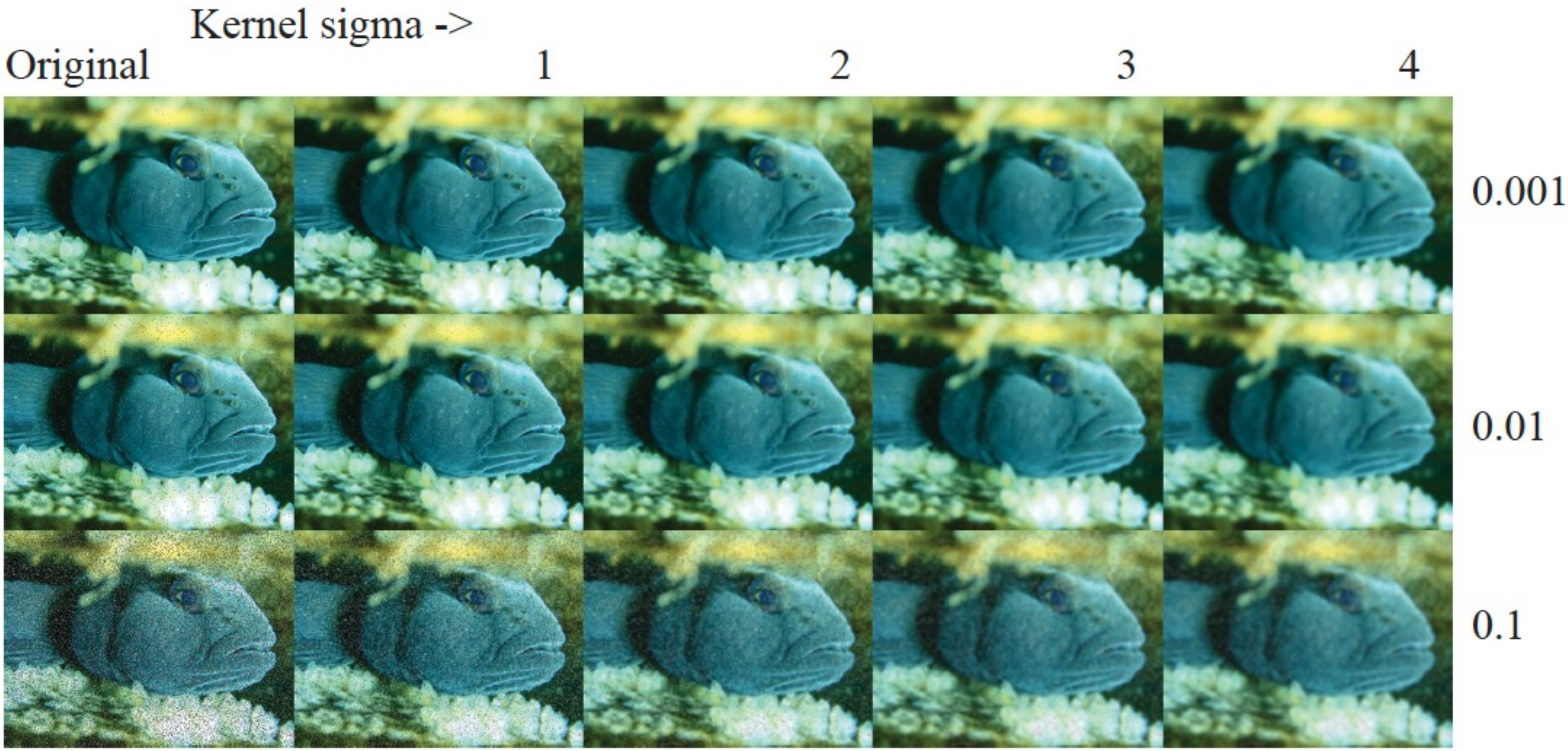
- If the standard deviation of the Gaussian is very small—say, smaller than one pixel—the smoothing will have little effect because the weights for all pixels off the center will be very small.
- For a larger standard deviation, the neighboring pixels will have larger weights in the weighted average, which in turn means that the average will be strongly biased toward a consensus of the neighbors. This will be a good estimate of a pixel's value, and the noise will largely disappear at the cost of some blurring.
- Finally, a kernel that has a large standard deviation will cause much of the image detail to disappear, along with the noise.

# Poisson noise

- For each pixel location, flip a biased coin
  - if it comes up heads, move on
  - if it comes up tails, flip a fair coin
    - if that is heads, pixel -> full bright
    - tails, pixel -> full dark

- Variants are possible
- Models device damage, manufacturing failures, some kinds of transmission error, etc.

# Smoothing Poisson noise with a gaussian filter

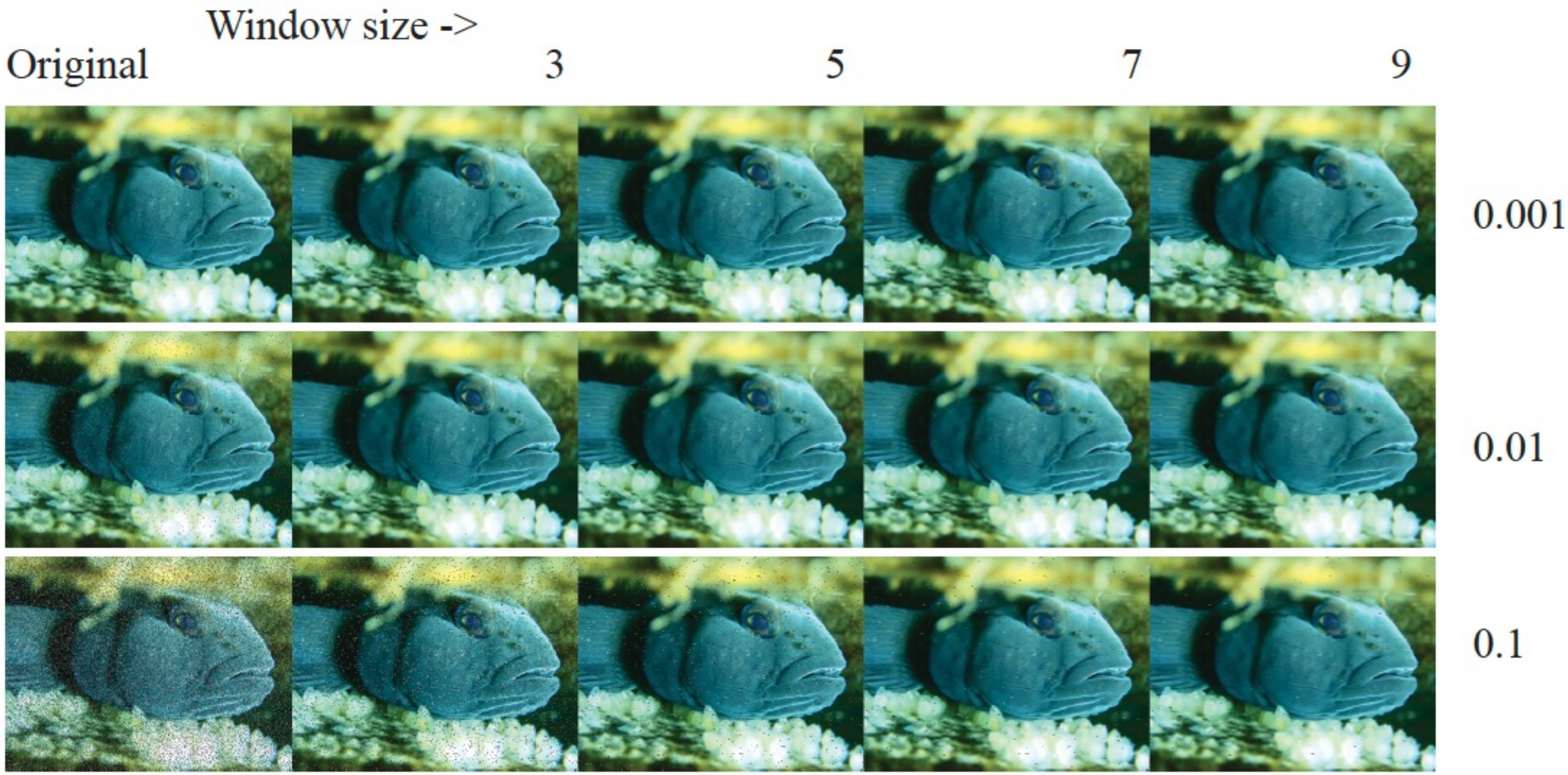


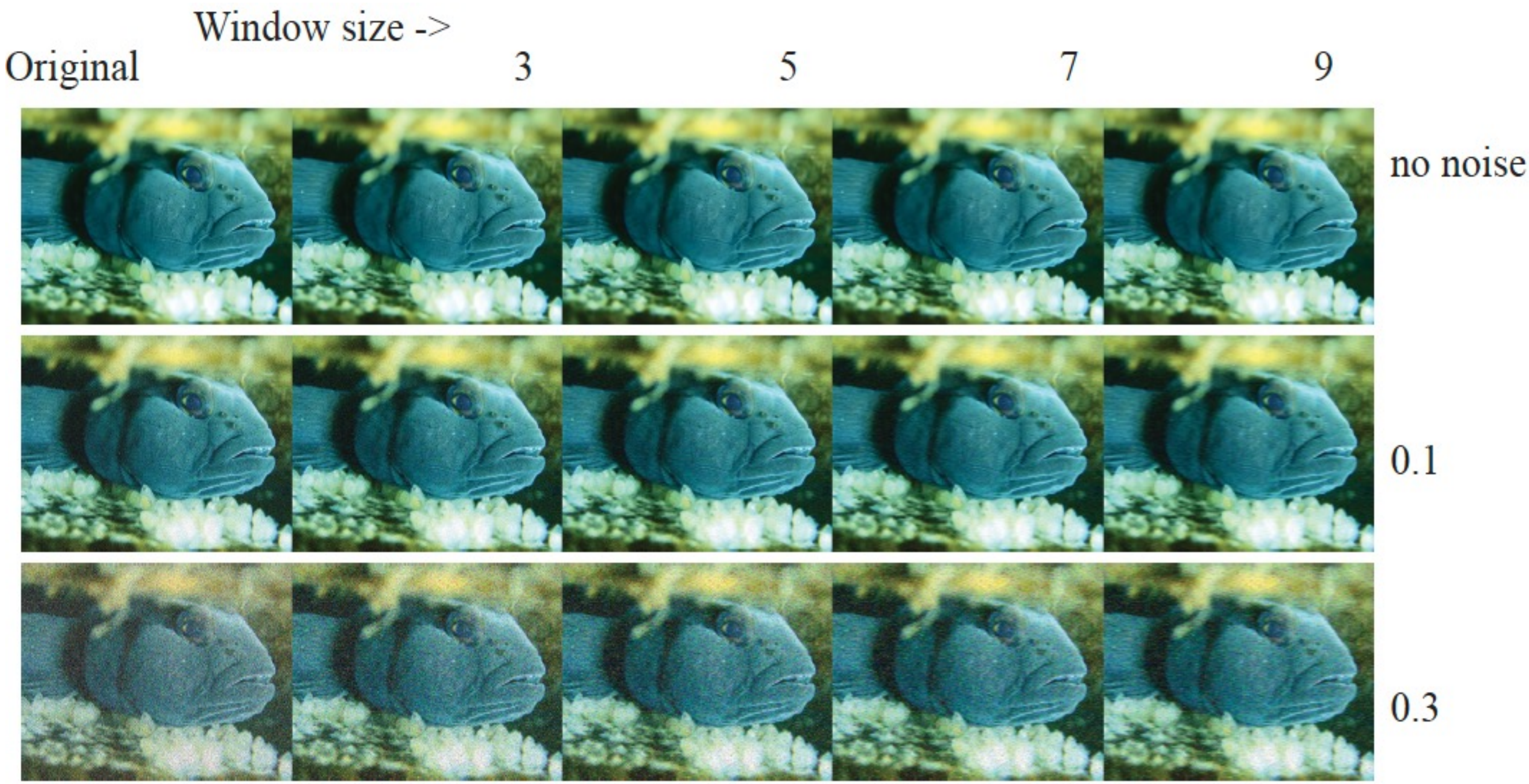
# The median filter

- $N_{ij} = \text{median}(\text{Neighborhood}(O_{ij}))$

- THIS ISN'T LINEAR!
  - (check you're sure of this)

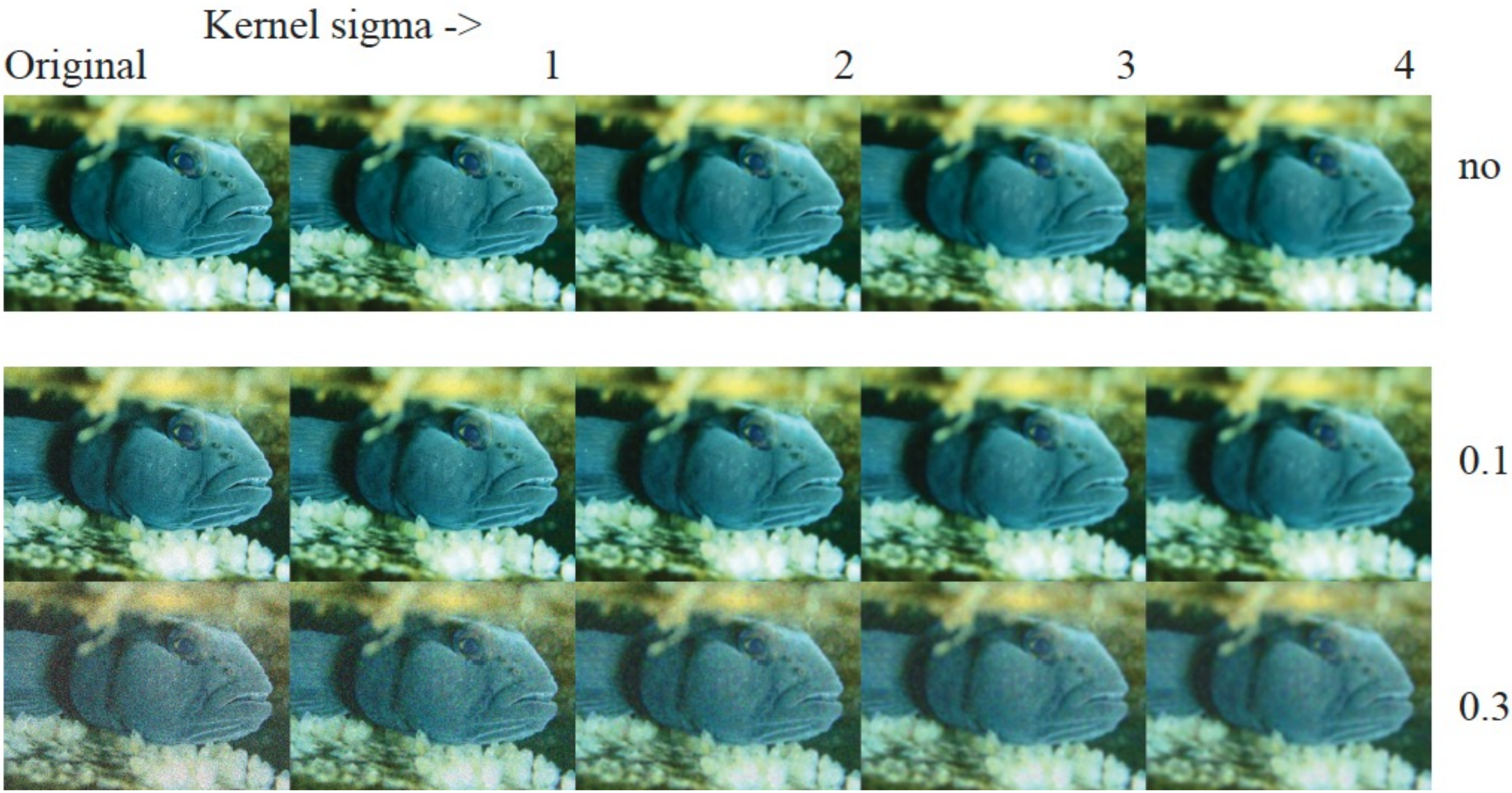
# Smoothing Poisson noise with a median filter





Window size ->
Original
3
5
7
9
no noise
0.1
0.3

# Gaussian smoothing of Gaussian noise



# Application: derivative of Gaussian filters

Because convolution is associative, smoothing an image and then differentiating it is the same as convolving it with the derivative of a smoothing kernel. First, differentiation is linear and shift invariant. This means that there is some kernel that differentiates. Given a function $I(x, y)$,

$$\frac{\partial I}{\partial x} = K_{(\partial/\partial x)} * I.$$

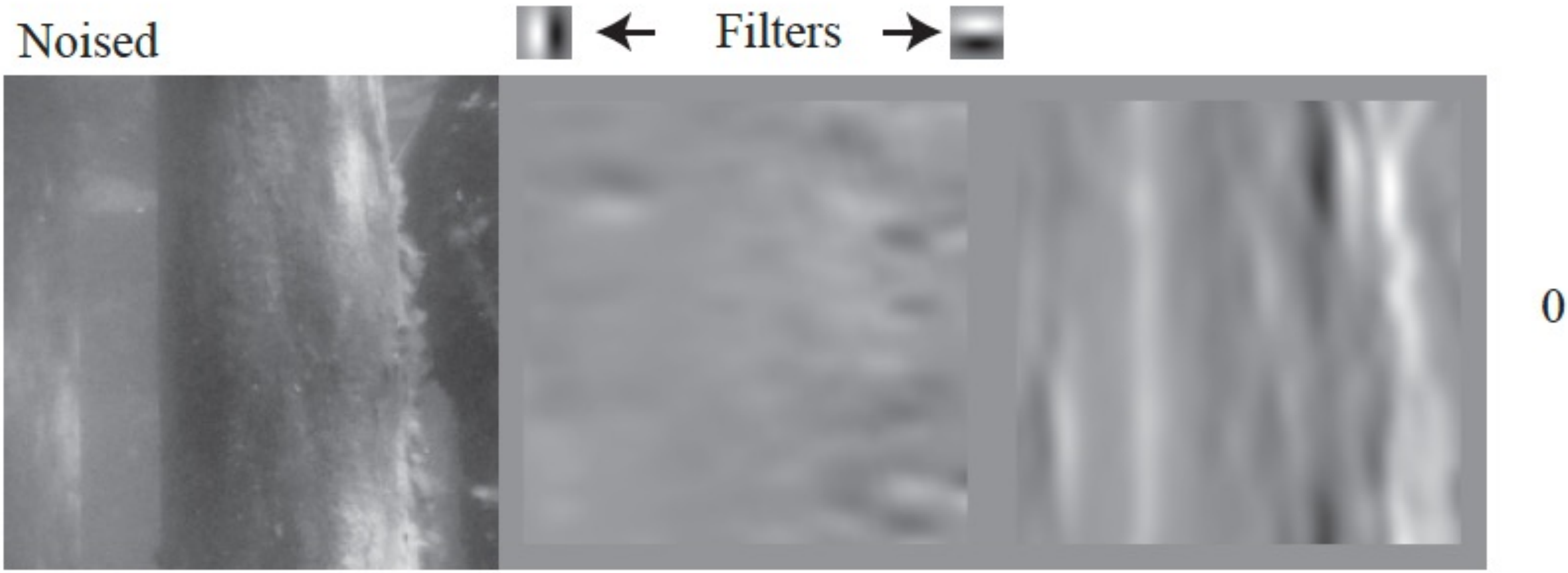
Write the convolution kernel for the smoothing as $S$. Now

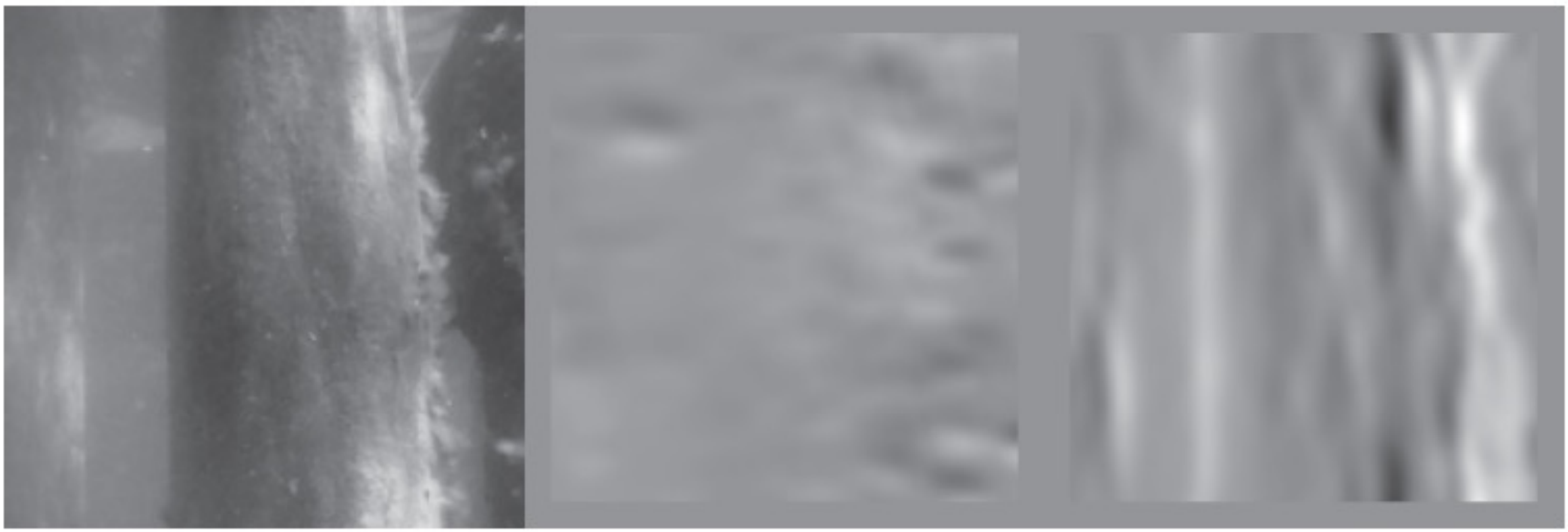
$$(K_{(\partial/\partial x)} * (S * I)) = (K_{(\partial/\partial x)} * S) * I = (\frac{\partial S}{\partial x}) * I.$$

Usually, the smoothing function is a gaussian, so an estimate of the derivative can be obtained by convolving with the derivative of the gaussian (rather than convolve and then differentiate), yielding

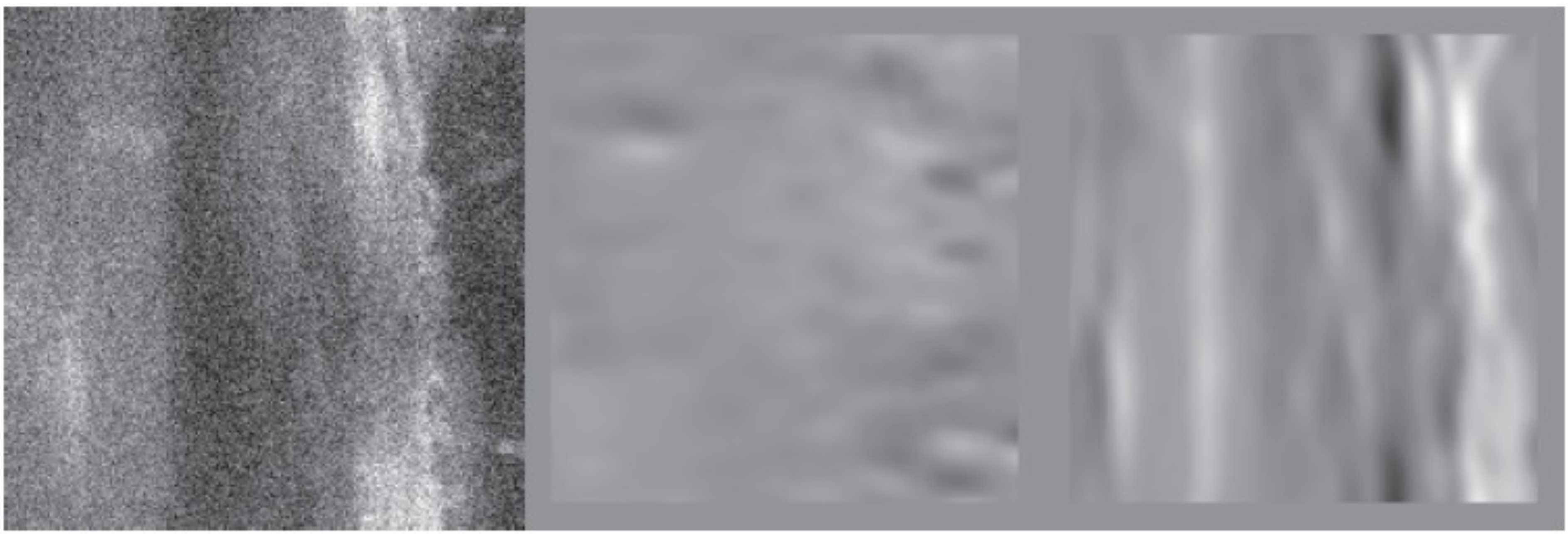
$$\begin{aligned} \frac{\partial g_\sigma}{\partial x} &= \frac{1}{2\pi\sigma^2}\left[\frac{-x}{2\sigma^2}\right] \exp -\left(\frac{x^2+y^2}{2\sigma^2}\right) \\ \frac{\partial g_\sigma}{\partial y} &= \frac{1}{2\pi\sigma^2}\left[\frac{-y}{2\sigma^2}\right] \exp -\left(\frac{x^2+y^2}{2\sigma^2}\right) \end{aligned}$$

SPOT THE TYPO!





0.01



0.1

# Think about this...

**6.13.** This exercise explores smoothing of additive Gaussian noise. Write $\mathcal{I}$ for an image whose $i$, $j$'th entry is $\mathcal{I}_{ij}$. Form $\mathcal{N}_{ij} = \mathcal{I}_{ij} + \sigma\xi_{ij}$, where $\xi_{ij}$ is an independent, identically distributed sample from a standard normal distribution (this has mean 0 and standard deviation 1). This means that, at each pixel in the image, you draw a sample from a standard normal distribution, scale it by $\sigma$, then add it to the pixel value. Write $\mathcal{K}$ for some $(2k+1) \times (2k+1)$ filter kernel, and $\Xi$ for the noise image (i.e. the image whose $i$, $j$'th component is $\xi_{ij}$).

**(a)** Form $\mathcal{M} = \mathcal{K} * \Xi$. Show that each pixel of this image is a sample of a normal distribution whose mean is 0 and whose standard deviation is $\sum_{ij} \mathcal{K}_{ij}^2$. What condition on $k$ ensures every pixel of $\mathcal{M}$ is independent of every other? For given $k$, characterize the pixels that are guaranteed to be independent of one another? What is the covariance of $\mathcal{M}_{ij}$ and $\mathcal{M}_{i+r,j+s}$ for given $r$ and $s$?

**(b)** Use the results of the previous exercise to argue that Gaussian smoothing suppresses Gaussian noise.