Convolution in more detail

- Some definitions and properties
- Application: Suppressing noise in derivatives
- Skate over:
  - Shift invariant + linear = convolution
- Application: Understanding interpolation and sampling

Some definitions

4.1.1 Convolution

For the moment, think of an image as a two dimensional array of intensities. Write $\mathcal{I}_{ij}$ for the pixel at position i, j. Construct a small array of weights (a *mask* or *kernel*) $\mathcal{W}$, and compute a new image $\mathcal{N}$ from the original image and the mask, using the rule

$$\mathcal{N}_{ij} = \sum_{uv} \mathcal{I}_{i-u,j-v} \mathcal{W}_{uv}$$

Discrete convolution

equivalently

$$\mathcal{N} = \mathcal{W} * \mathcal{I}.$$

In these slides, I really am talking about convolution (not correlation or filtering – matters for unit impulse)

Some definitions

$$\int\int g(x-x',y-y')h(x',y')dxdy$$ Continous convolution

$$(g*h)(x,y),$$

In these slides, I really am talking about convolution (not correlation or filtering – matters for unit impulse)

Some definitions

Most imaging systems have, to a good approximation, three significant properties. Write $R(f)$ for the response of the system to input f. Then the properties are:

- **Superposition:** the response to the sum of stimuli is the sum of the individual responses, so
$$R(f+g) = R(f) + R(g);$$
- **Scaling:** the response to a scaled stimulus is a scaled version of the response to the original stimulus, so
$$R(kf) = kR(f).$$
An operation that exihibits superposition and scaling is *linear*.
- **Shift invariance:** In a shift invariant linear system, the response to a translated stimulus is just a translation of the response to the stimulus. This means that, for example, if a view of a small light aimed at the center of the camera is a small, bright blob, then if the light is moved to the periphery, the response is same small, bright blob, only translated.

A device that is linear and shift invariant is known as a *shift invariant linear system*. The operation represented by the device is a *shift invariant linear operation*.

Example: a lens

- Superposition: yes
- Scaling: yes
- So linear

- Shift invariant: yes
    - (ish – think about the edges of the field of view)

Properties

Convolution is:

- linear, by construction;
- shift-invariant, by construction;
- *commutative* (meaning

$$(g * h)(x) = (h * g)(x)$$

  exercises);

- *associative* (meaning that

$$(f * (g * h)) = ((f * g) * h)$$

  exercises).

Applications: Gradient estimates

For an image $\mathcal{I}$, the gradient is

$$\nabla \mathcal{I} = \left(\frac{\partial \mathcal{I}}{\partial x}, \frac{\partial \mathcal{I}}{\partial y}\right)^T,$$

which we could estimate by observing that

$$\frac{\partial \mathcal{I}}{\partial x} = \lim_{\delta x \to 0} \frac{\mathcal{I}(x + \delta x, y) - \mathcal{I}(x, y)}{\delta x} \approx \mathcal{I}_{i+1,j} - \mathcal{I}_{i,j}.$$

This means a convolution with

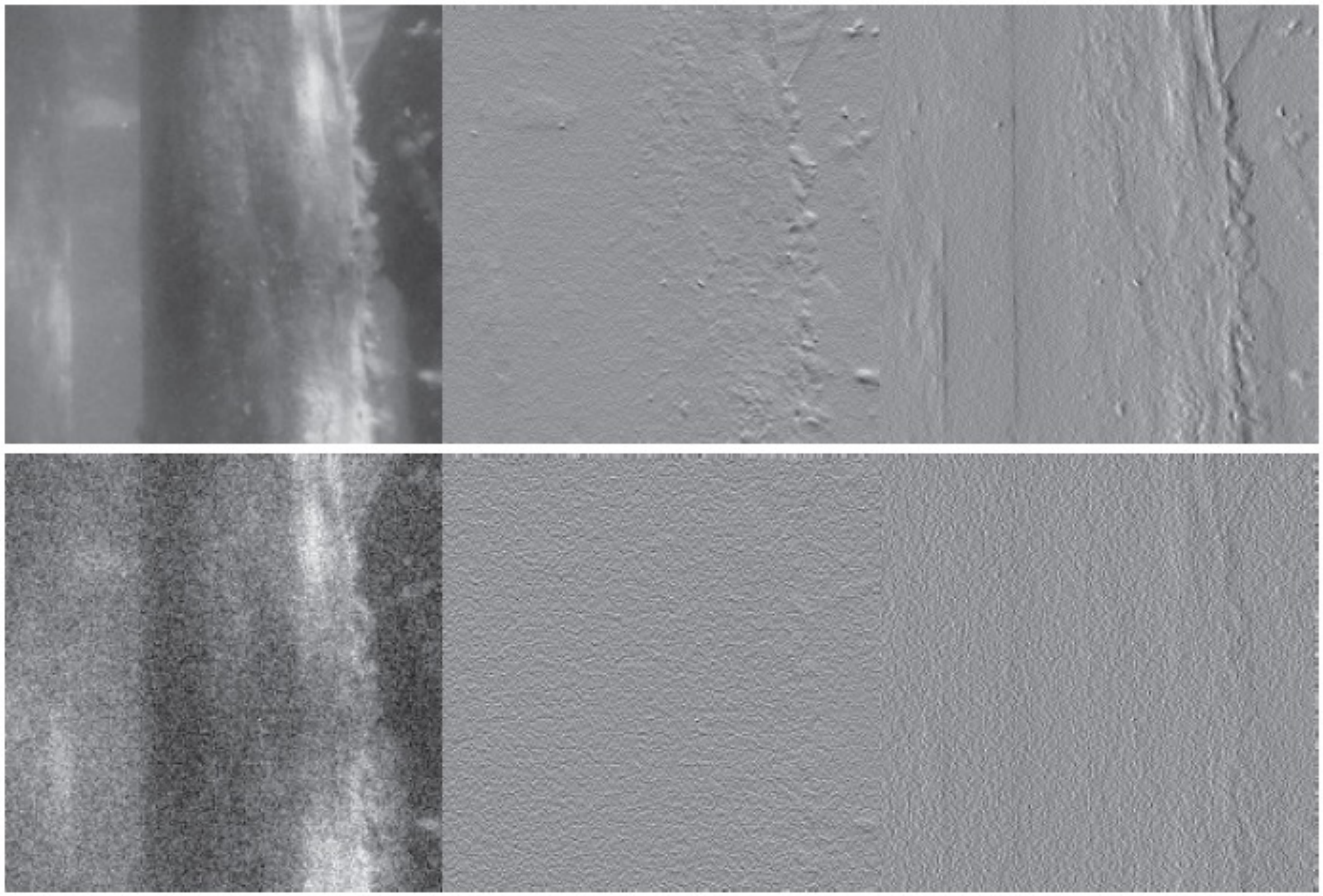
$$-1 \quad 1$$

will estimate $\partial \mathcal{I}/\partial x$ (nothing in the definition requires convolution with a square kernel). Notice that this kernel "looks like" a dark pixel next to a light pixel, and will respond most strongly to that pattern. By the same argument, $\partial \mathcal{I}/\partial y \approx \mathcal{I}_{i,j+1} - \mathcal{I}_{i,j}$. These kinds of derivative estimates are known as *finite differences*.

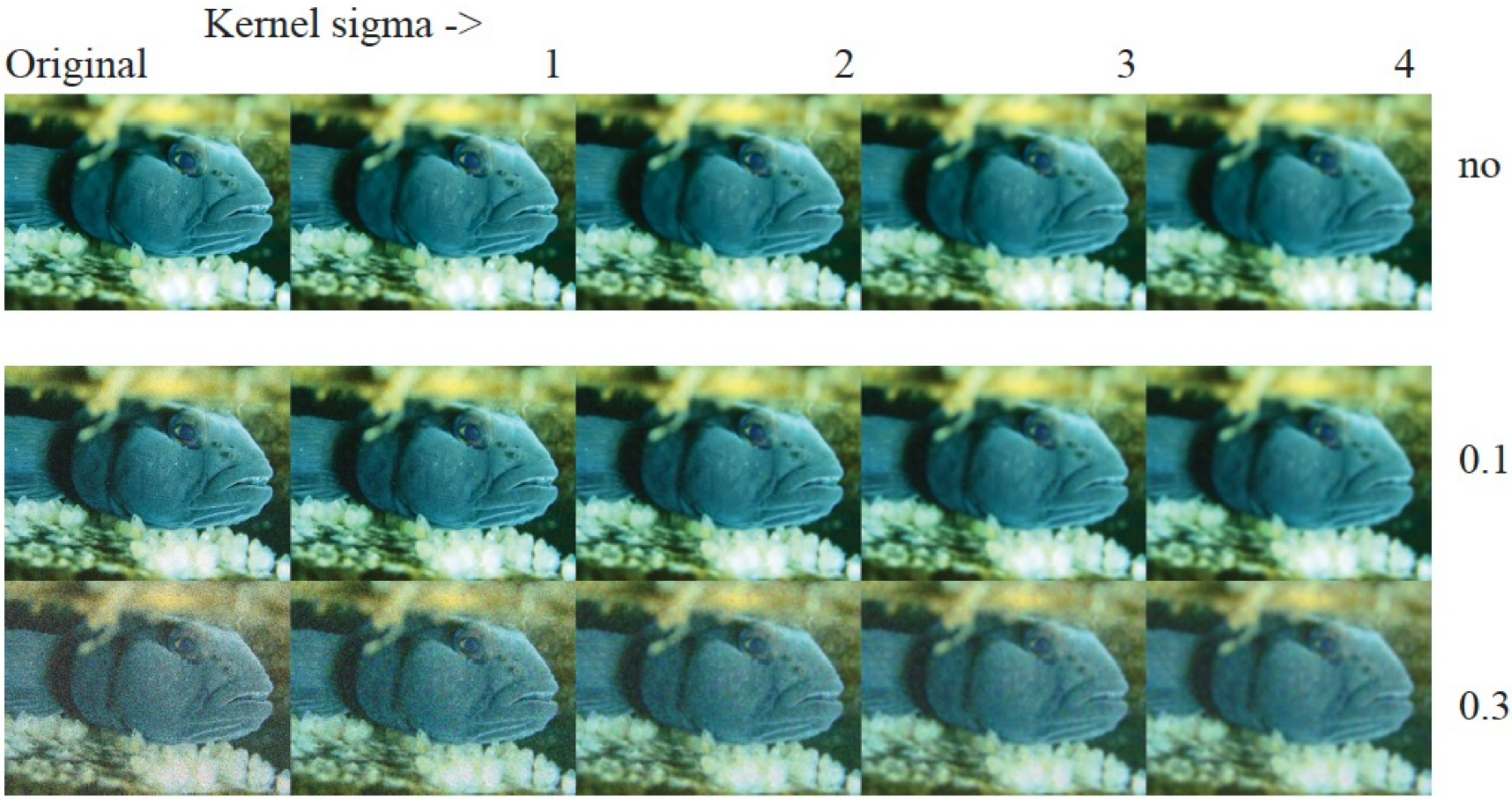
Finite differences are overexcited by noise

0.01

0.1



Gaussian smoothing of Gaussian noise



So…

- Suppress noise by smoothing image
- Then convolve with derivative
- Two convolutions – but convolution is associative

Application: Derivative of Gaussian Filters

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

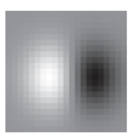
$$(K_{(\partial/\partial x)} * (S * I)) = (K_{(\partial/\partial x)} * S) * I = (\frac{\partial S}{\partial x}) * I.$$

Derivative of Gaussian Filters

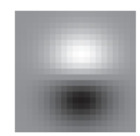
Usually, the smoothing function is a gaussian, so an estimate of the derivative can be obtained by convolving with the derivative of the gaussian (rather than convolve and then differentiate), yielding

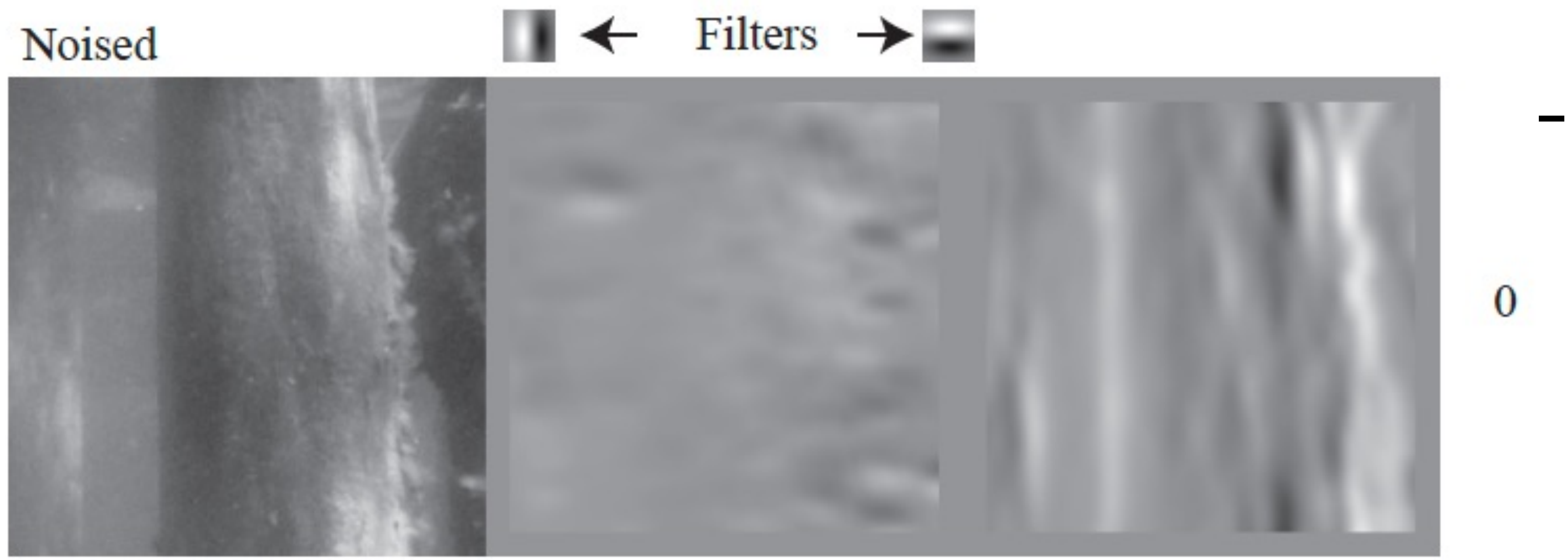
$$\frac{\partial g_\sigma}{\partial x} = \frac{1}{2\pi\sigma^2}\left[\frac{-x}{2\sigma^2}\right]\exp - \left(\frac{x^2+y^2}{2\sigma^2}\right)$$

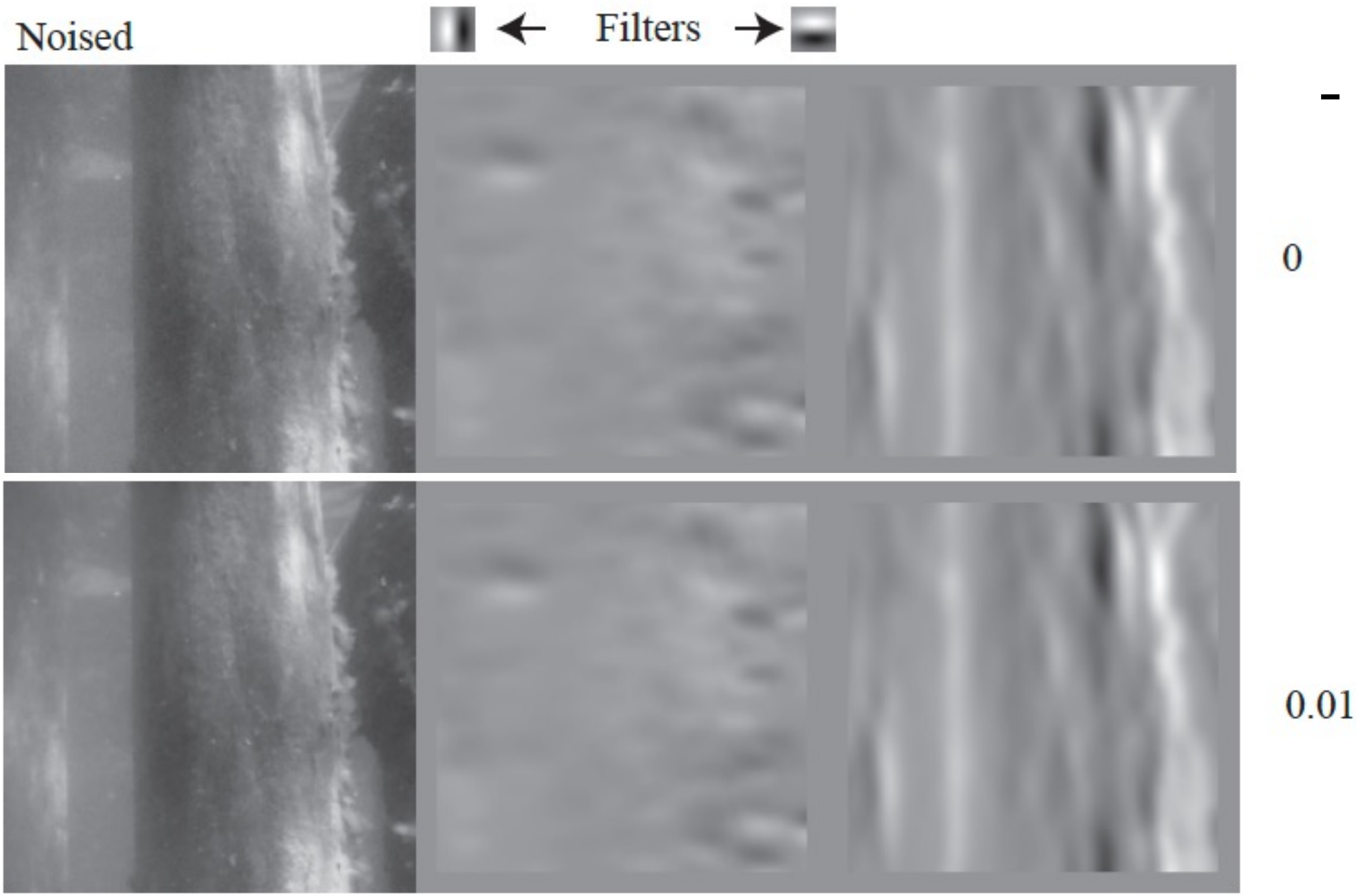
$$\frac{\partial g_\sigma}{\partial y} = \frac{1}{2\pi\sigma^2}\left[\frac{-y}{2\sigma^2}\right]\exp - \left(\frac{x^2+y^2}{2\sigma^2}\right)$$



← Filters →

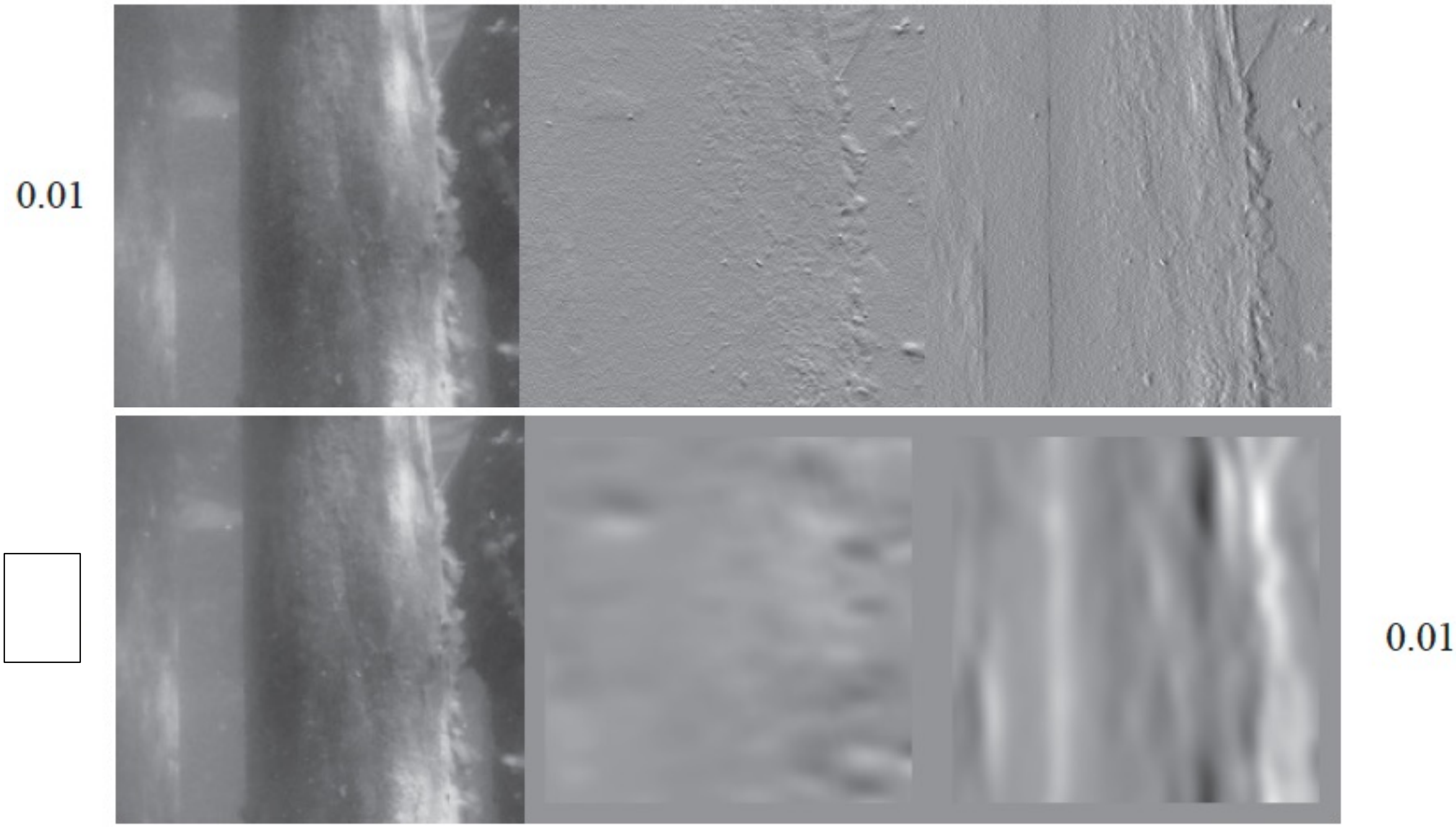


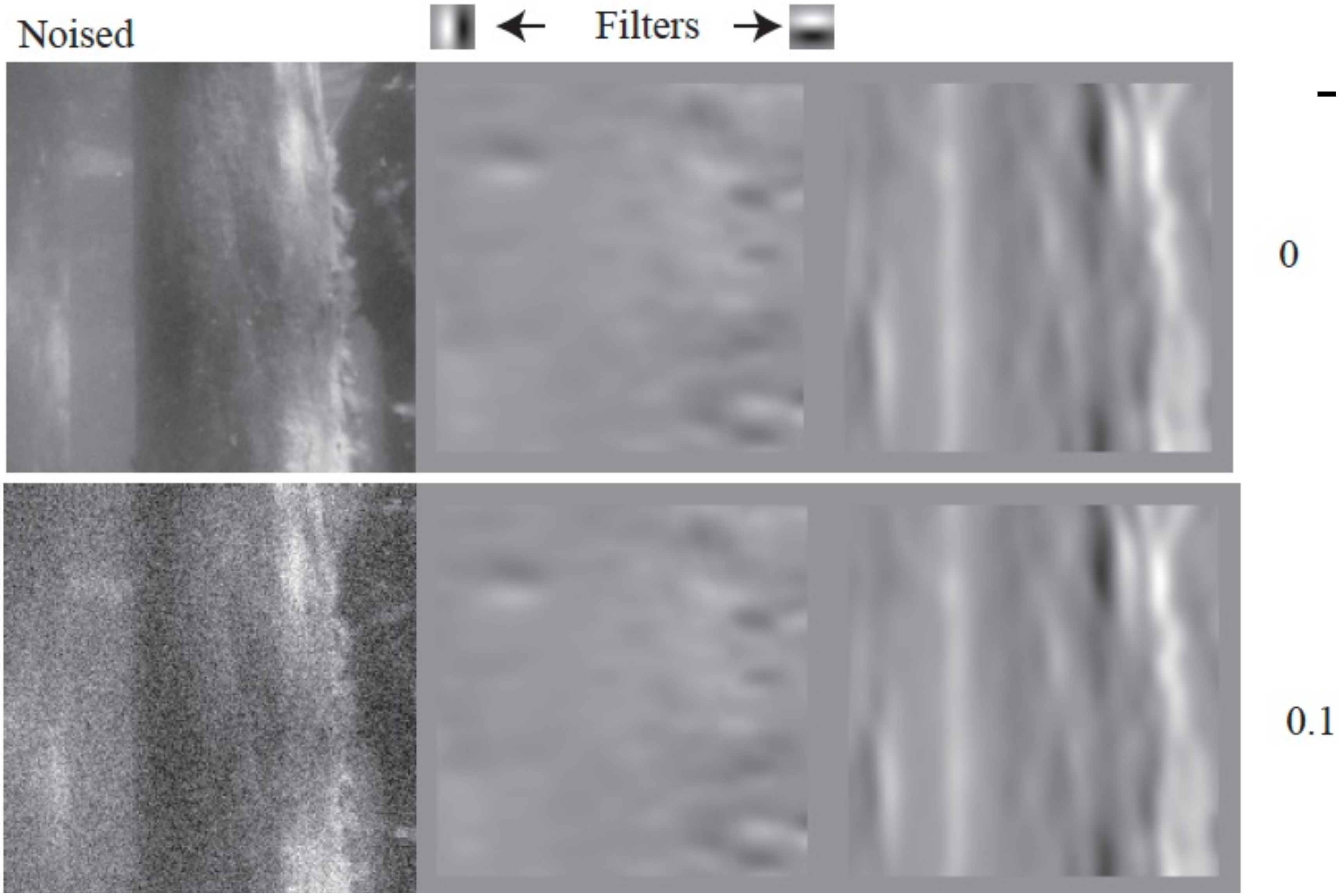

Noised
Filters
0



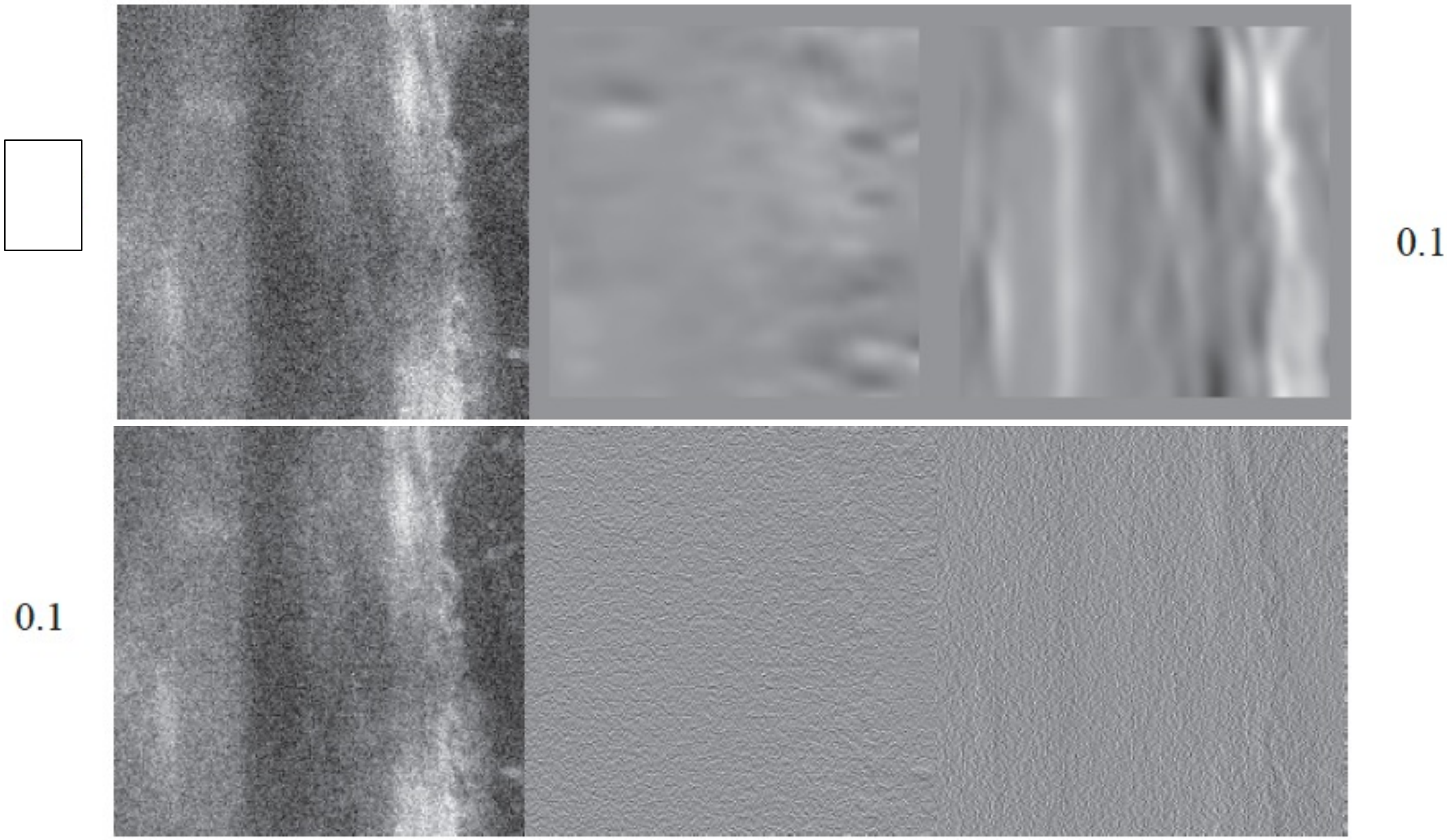
Noised
Filters
-
0
0.01

DOG filters are not excited by noise




Noised
Filters
-
0
0.1

DOG filters are not excited by noise



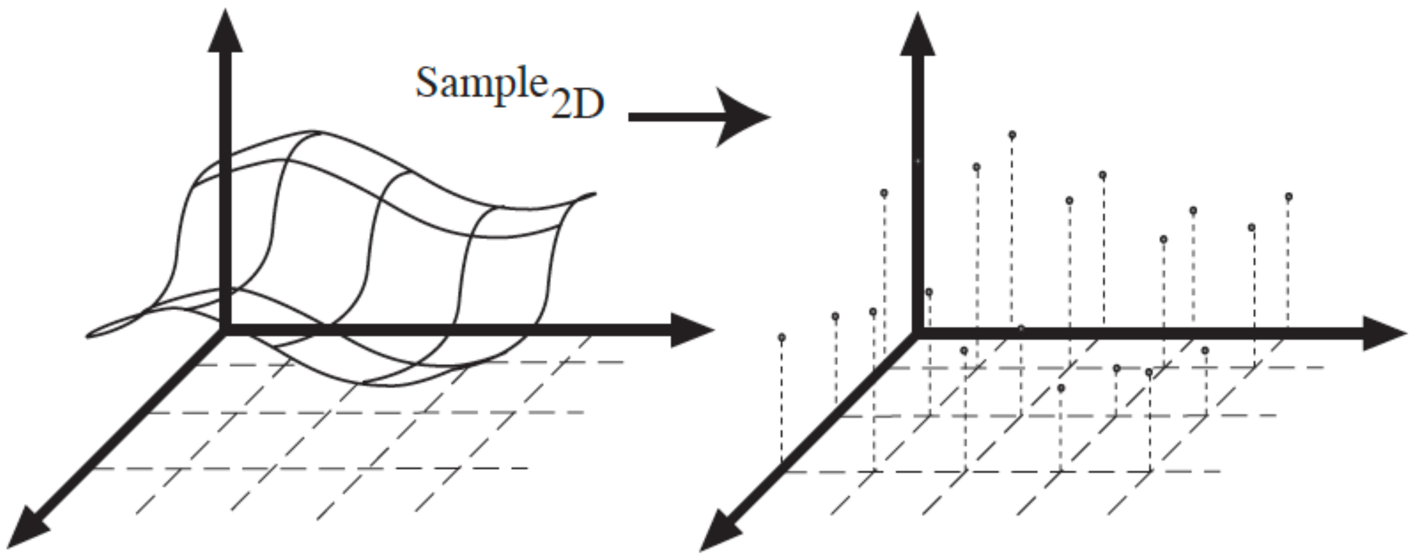
Important facts

- A shift invariant linear operator is equivalent to a convolution

  - In all cases
    - Discrete 1D
    - Discrete 2D
    - Discrete ND
    - Continuous 1D
    - Continuous 2D
    - Continuous ND

- Interesting thought experiment:
  - Differentiation in 1D is linear and shift invariant – what is the convolution kernel
  - Discrete – easy
  - Continuous – bit weird, but fairly easy

Sampling



But…

How to represent the sampled signal as a continuous function?

Understanding aliasing will require a continuous model of a sampled signal. Write $C(\mathcal{I})$ for the operation that maps a sampled image $\mathcal{I}$ to this continuous model. This model should respect convolution and sampling in a sensible way. Choose some continuous convolution kernel $g(x, y)$ A desirable property of this model is that if you convolve $C(\mathcal{I})$ with $g(x, y)$, then sample the result, you get what you would have gotten if you convolve $\mathcal{I}$ with $\texttt{sample}_{2D}(g)$. To write this out, it is helpful to distinguish discrete convolution (I will write $*_d$) and continuous convolution (I will write $*_c$). The property is:

Continuous convolution

Discrete convolution

$$\texttt{sample}_{2D}(\underline{C(\mathcal{I})} *_c g) = \mathcal{I} *_d \texttt{sample}_{2D}(g).$$

Continuous model of sampled image

Discrete image

The delta function

box function is now given by $box_{\epsilon^2}(x, y) = box_\epsilon(x) box_\epsilon(y)$ and

$$d_\epsilon(x, y) = \frac{box_{\epsilon^2}(x, y)}{\epsilon^2}.$$

The δ-function is the limit of $d_\epsilon(x, y)$ function as $\epsilon \to 0$. Again, discussion of the value of $\delta(0)$ is better avoided, but notice

$$\int_{-\infty}^{\infty} g(x, y) \delta(x, y) dx dy = g(0, 0).$$

The right continuous model of a sampled signal

Now $C(\mathcal{I}$ cannot just be a function that takes the value of the signal at integer points and is zero everywhere else, because this model has a zero integral so the left hand side will be zero. Instead, use

$$C(\mathcal{I})(x, y) = \sum_{i,j} \mathcal{I}_{ij} \delta(x - i, y - j)$$

and find

$$C(\mathcal{I}) *_c g = \sum_{i,j} \mathcal{I}_{ij} g(x - x_i, y - y_j)$$

so that the u, v'th component of

$$\texttt{sample}_{2D}(C(\mathcal{I}) *_c g) \text{ is } \sum_{i,j} \mathcal{I}_{ij} g(x_u - x_i, y_v - y_j)$$

and the property holds.

Interpolation

Recall the interpolate of Section 2.2 had the form

$$\mathcal{I}(x, y) = \sum \mathcal{I}_{ij} b(x - i, y - j).$$

b is some function with the properties $b(0, 0) = 1$ and $b(u, v) = 0$ for u and v any other grid point. This is linear and shift invariant (exercises) so it must be a convolution. The way to see the convolution is to use the continuous model of the sampled image. This exposes the convolution in interpolation. Notice that

$$\begin{aligned} C(\mathcal{I}) * b &= \int\int C(\mathcal{I})(x - u, y - v) b(u, v) du dv \\ &= \sum_{i,j} \mathcal{I}_{ij} \int\int \delta(x - u - i, y - v - j) b(u, v) du dv \\ &= \sum_{i,j} \mathcal{I}_{ij} b(x - i, y - j) \text{ from the property of a } \delta \text{ function} \end{aligned}$$

which is the form of an interpolate.