Convolution

NEU 466M

Spring 2020

Convolution

$$\{\cdots g_{t-1}, g_t, g_{t+1} \cdots\}$$

g: a time-varying signal sampled at discrete intervals

$$\{\cdots h_{t-1}, h_t, h_{t+1} \cdots\}$$

h: another time-varying signal, not necessarily of same length

$$(g * h)(n) = \sum_{m=-\infty}^{\infty} g(n-m)h(m)$$

Finite-length g, h: g*h has length N+M-1, where N=length(g), M=length(h).

Properties of convolution

$$(g * h)(n) = \sum_{m=-\infty}^{\infty} g(n-m)h(m)$$

- Commutative/symmetric (unlike cross-correlation): $g * h = h * g$

- Associative: $f * (g * h) = (f * g) * h$

- Distributive: $f * (g + h) = f * g + f * h$

Convolution

$$(g * h)(n) = \sum_{m=-\infty}^{\infty} g(n-m)h(m)$$

- Typically, one short series, one long.
- Long series called “signal”; the other called the “kernel”.
- The convolution is viewed as a weighted version/moving average of the by the kernel.

Convolution

$$(g * h)(n) = \sum_{m=-\infty}^{\infty} g(n-m)h(m)$$

Say h: kernel (short/"finite support"), g: signal (long).

$$[\cdots g_{n-3} \quad g_{n-2} \quad g_{n-1} \quad g_n \quad g_{n+1} \quad g_{n+2} \quad g_{n+3} \cdots]$$

$$[\cdots \quad 0 \quad h_2 \quad h_1 \quad h_0 \quad h_{-1} \quad h_{-2} \quad 0 \quad \cdots]$$

$$(g * h)_n$$

Convolution

$$(g * h)(n) = \sum_{m=-\infty}^{\infty} g(n-m)h(m)$$

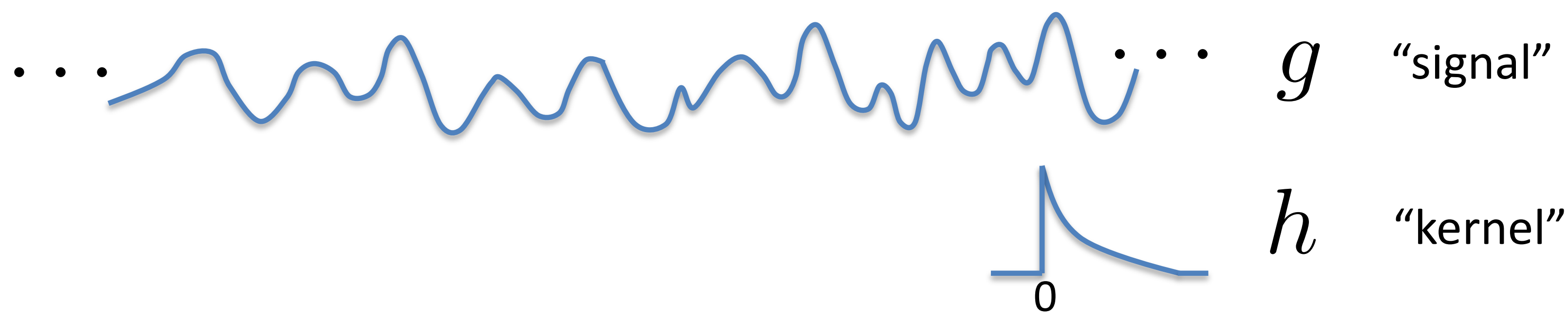
$$[\cdots \quad g_{n-2} \quad g_{n-1} \quad g_n \quad g_{n+1} \quad g_{n+2} \quad g_{n+3} \quad g_{n+4} \cdots]$$

$$[\cdots \quad 0 \quad h_2 \quad h_1 \quad h_0 \quad h_{-1} \quad h_{-2} \quad 0 \quad \cdots]$$

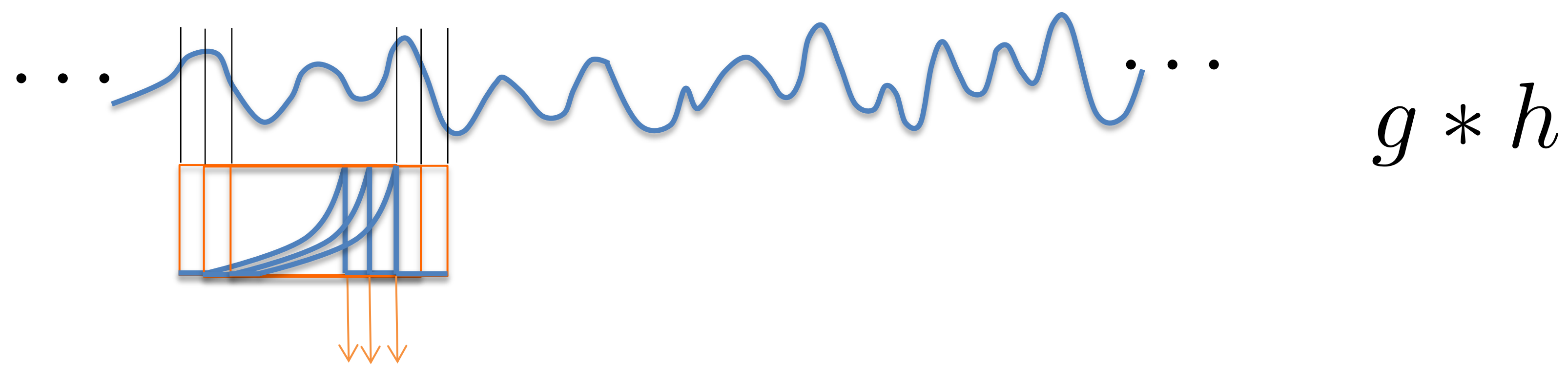
$$\underbrace{h_2 \quad h_1 \quad h_0 \quad h_{-1} \quad h_{-2}}_{(g * h)_{n+1}}$$

Flip *h* (kernel) and keep it in one place, move *g*-tape (signal) left.

Convolution

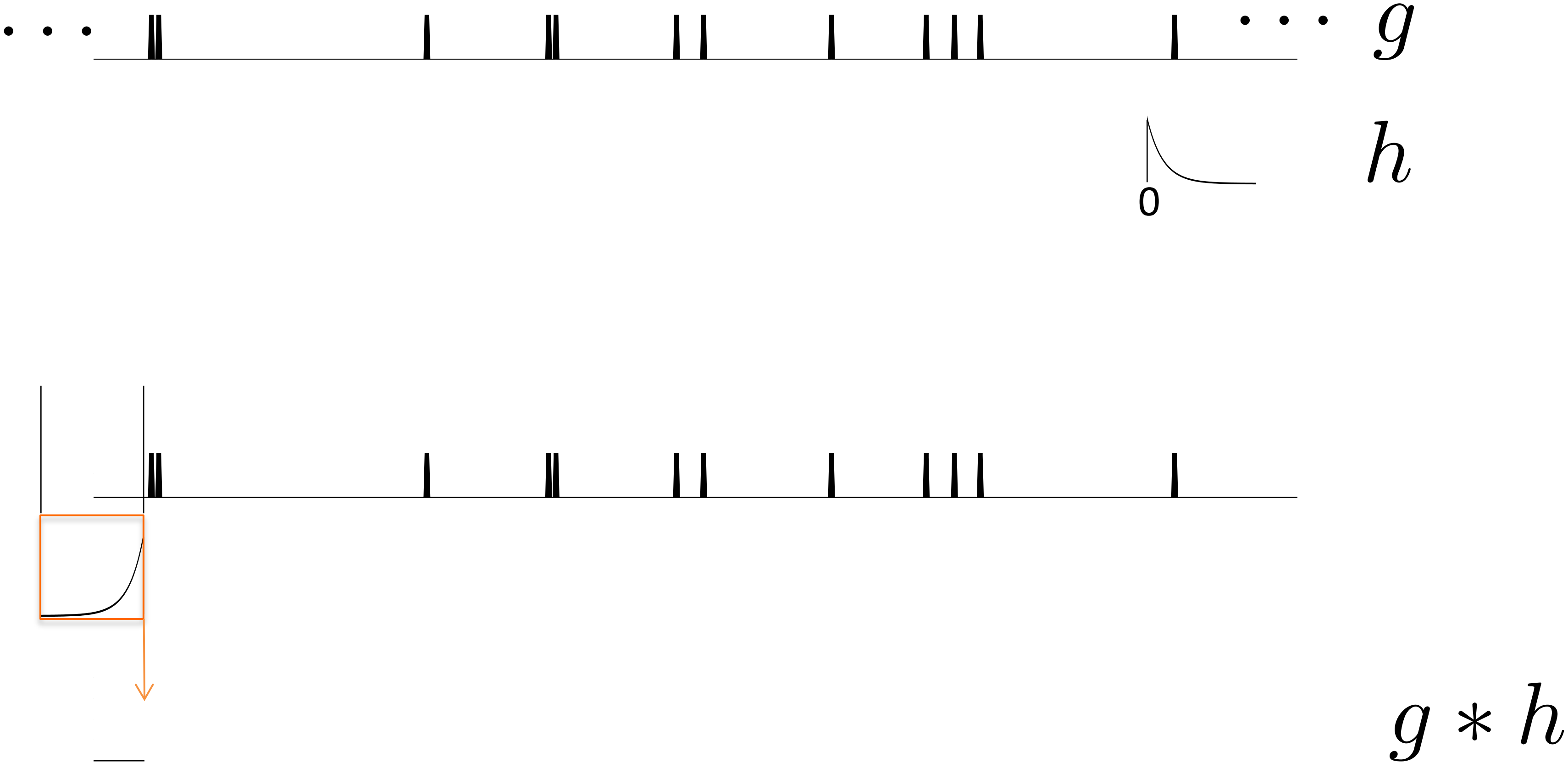


$$(g * h)(n) = \sum_{m=-\infty}^{\infty} g(n-m)h(m)$$

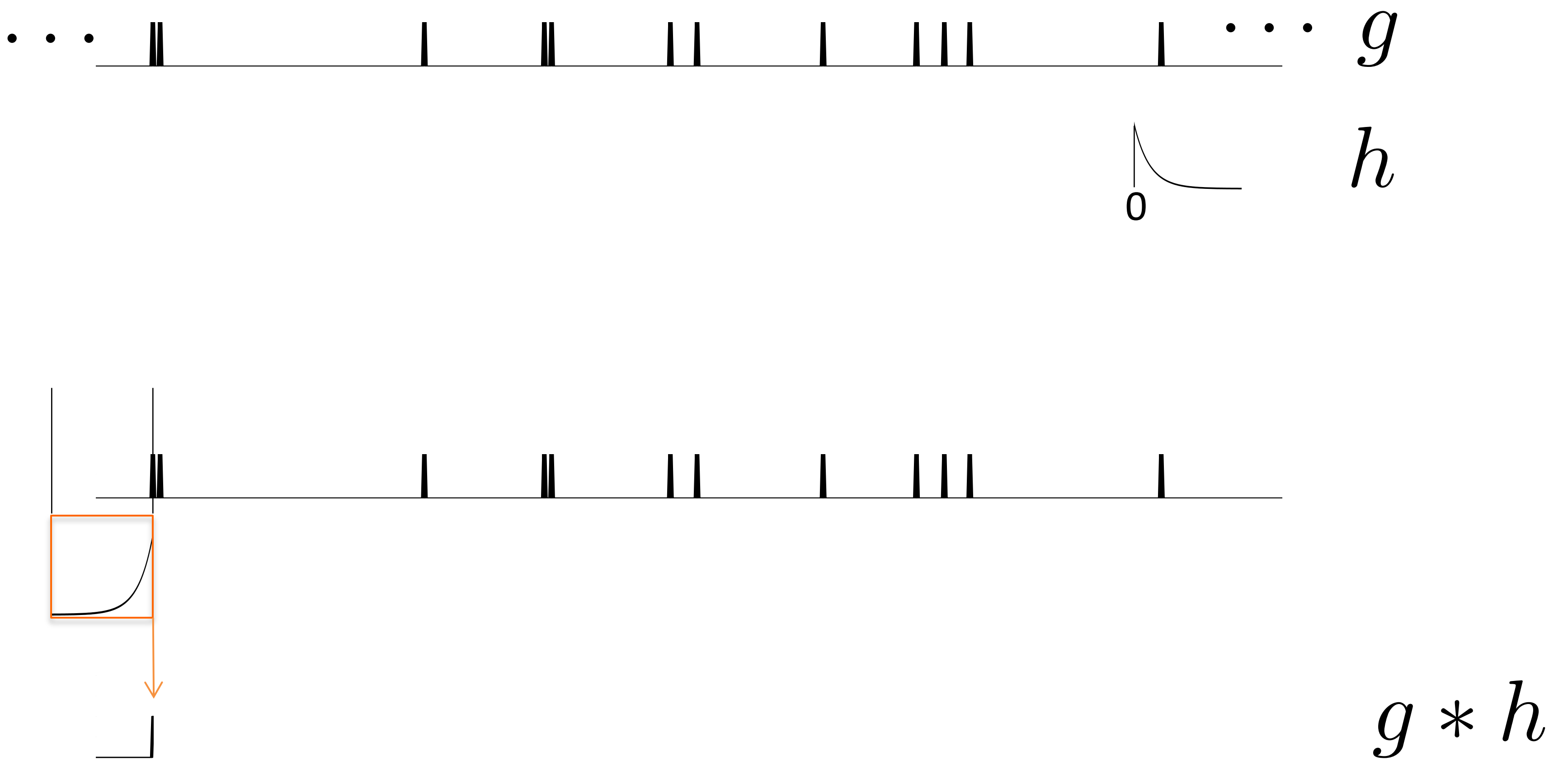


Flip *h* (kernel), keep *g*-tape (signal) fixed, sweep *h* (kernel) rightward.

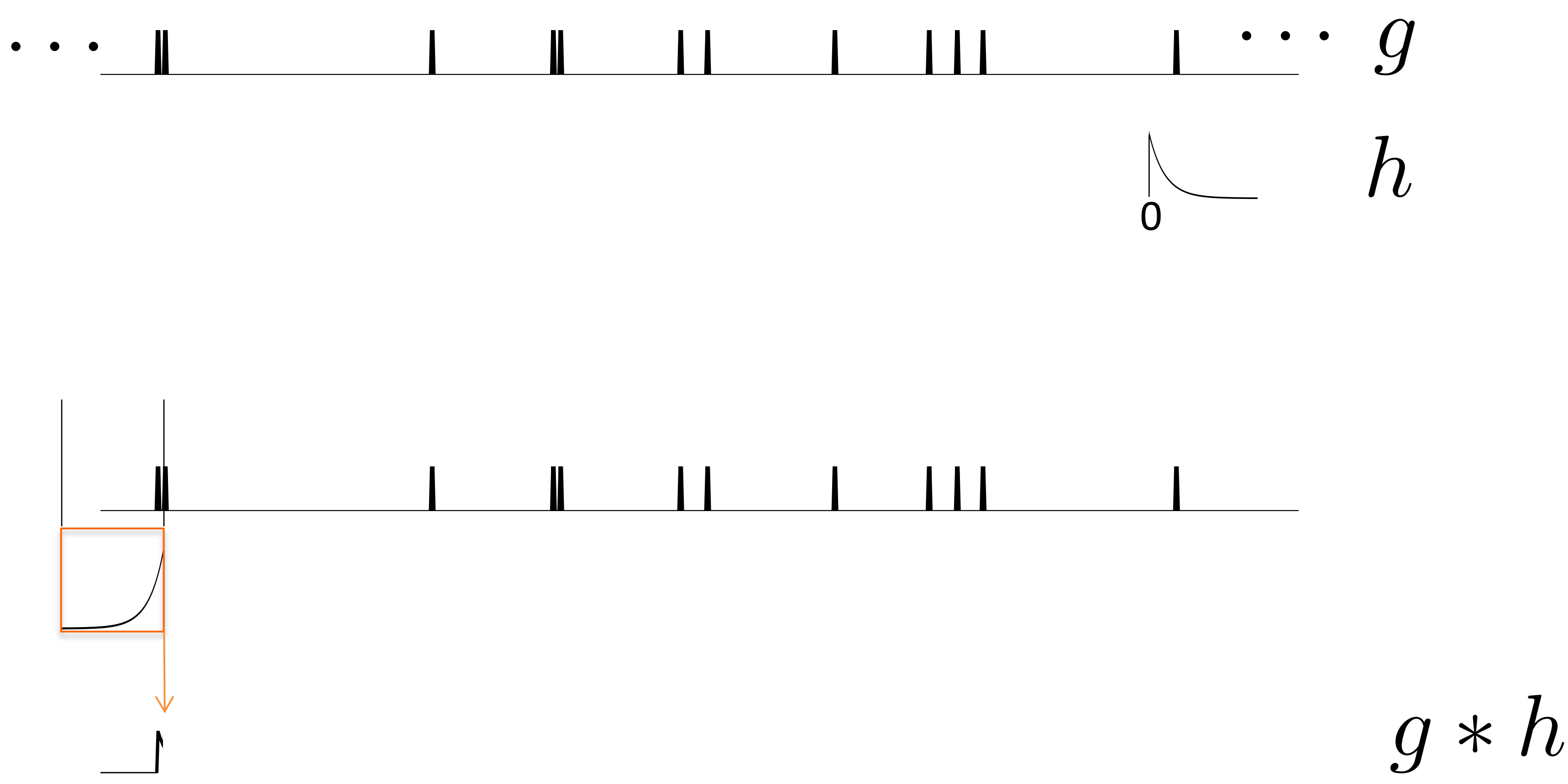
Convolution



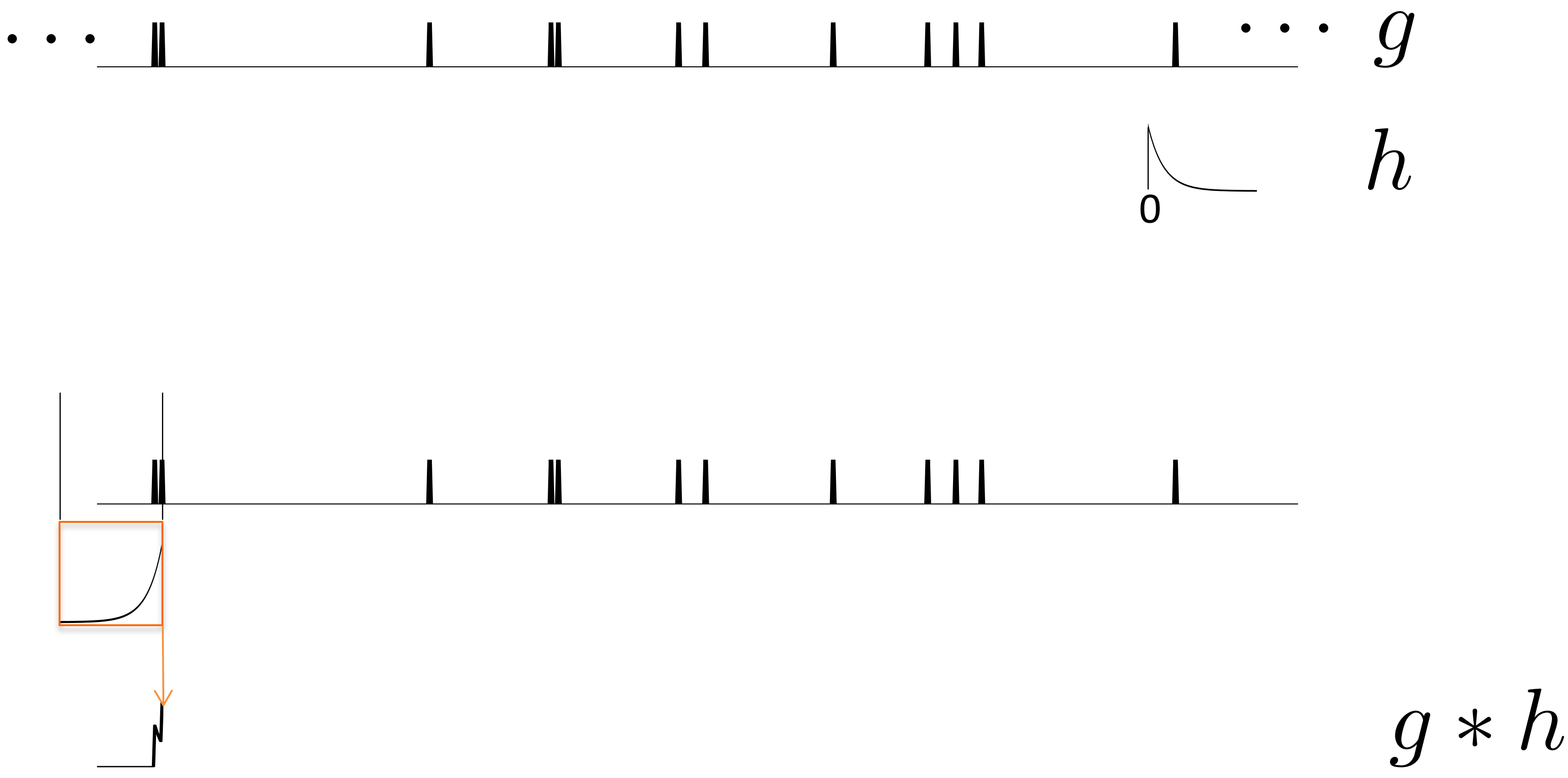
Convolution



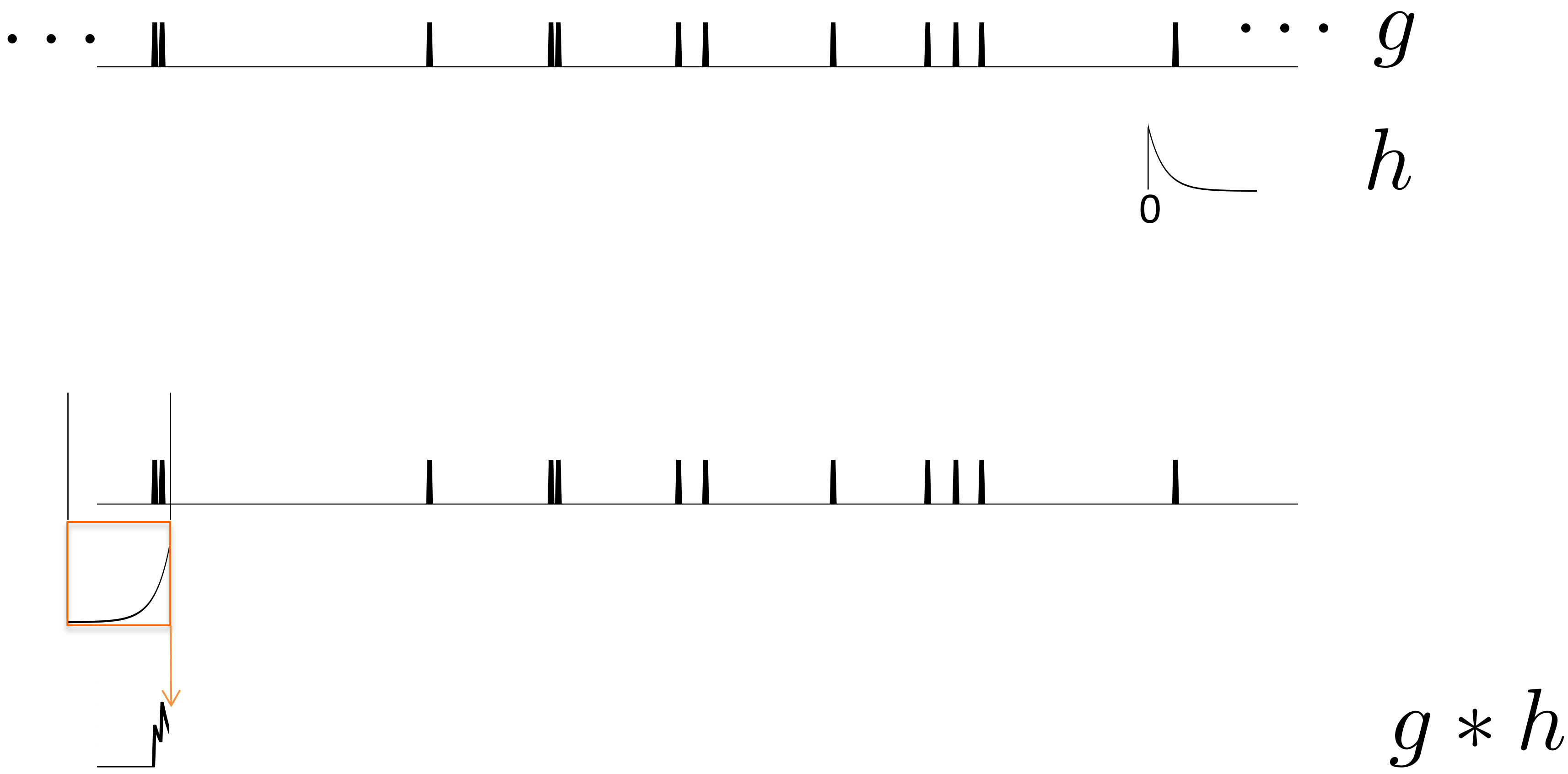
Convolution



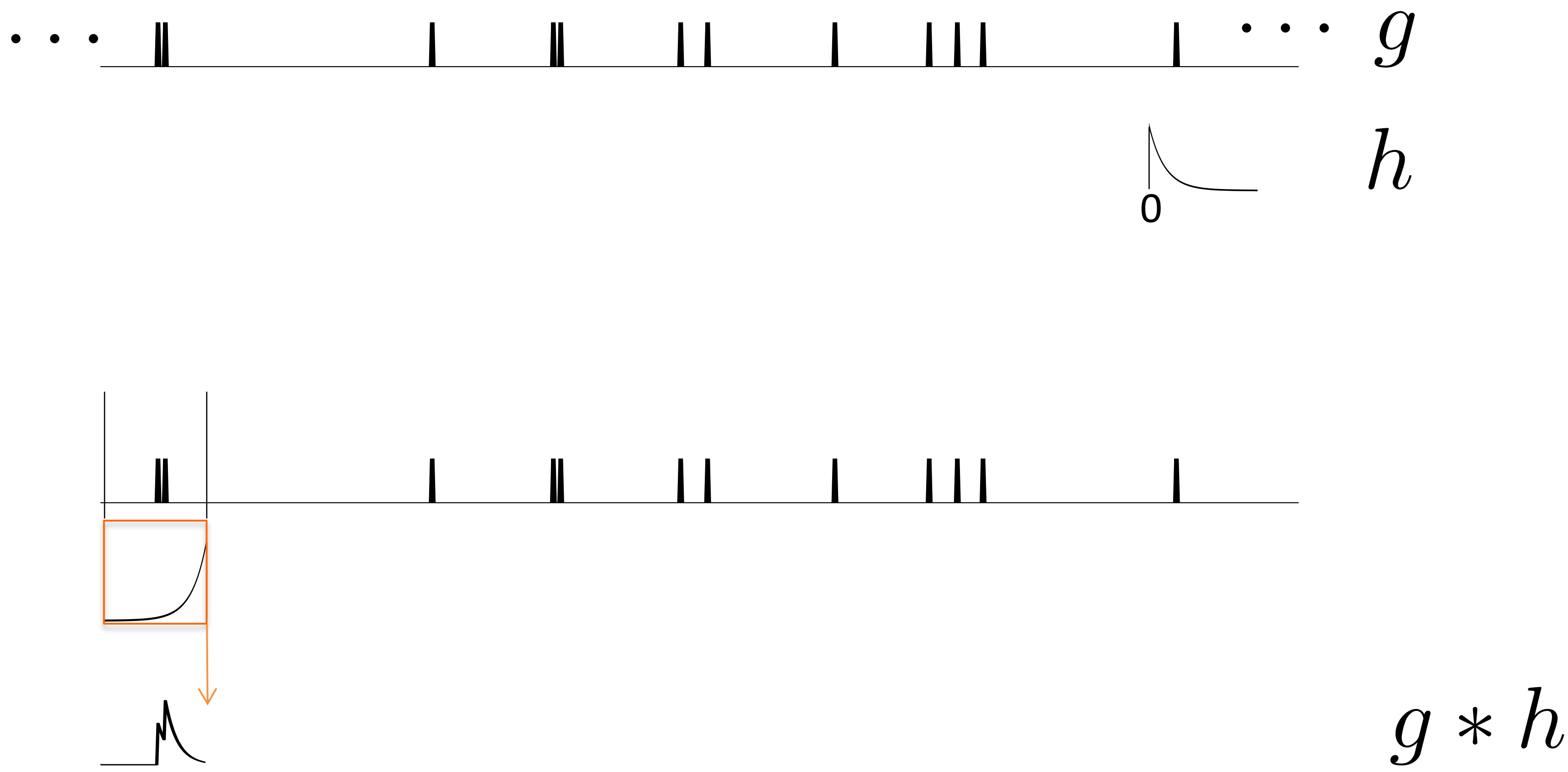
Convolution



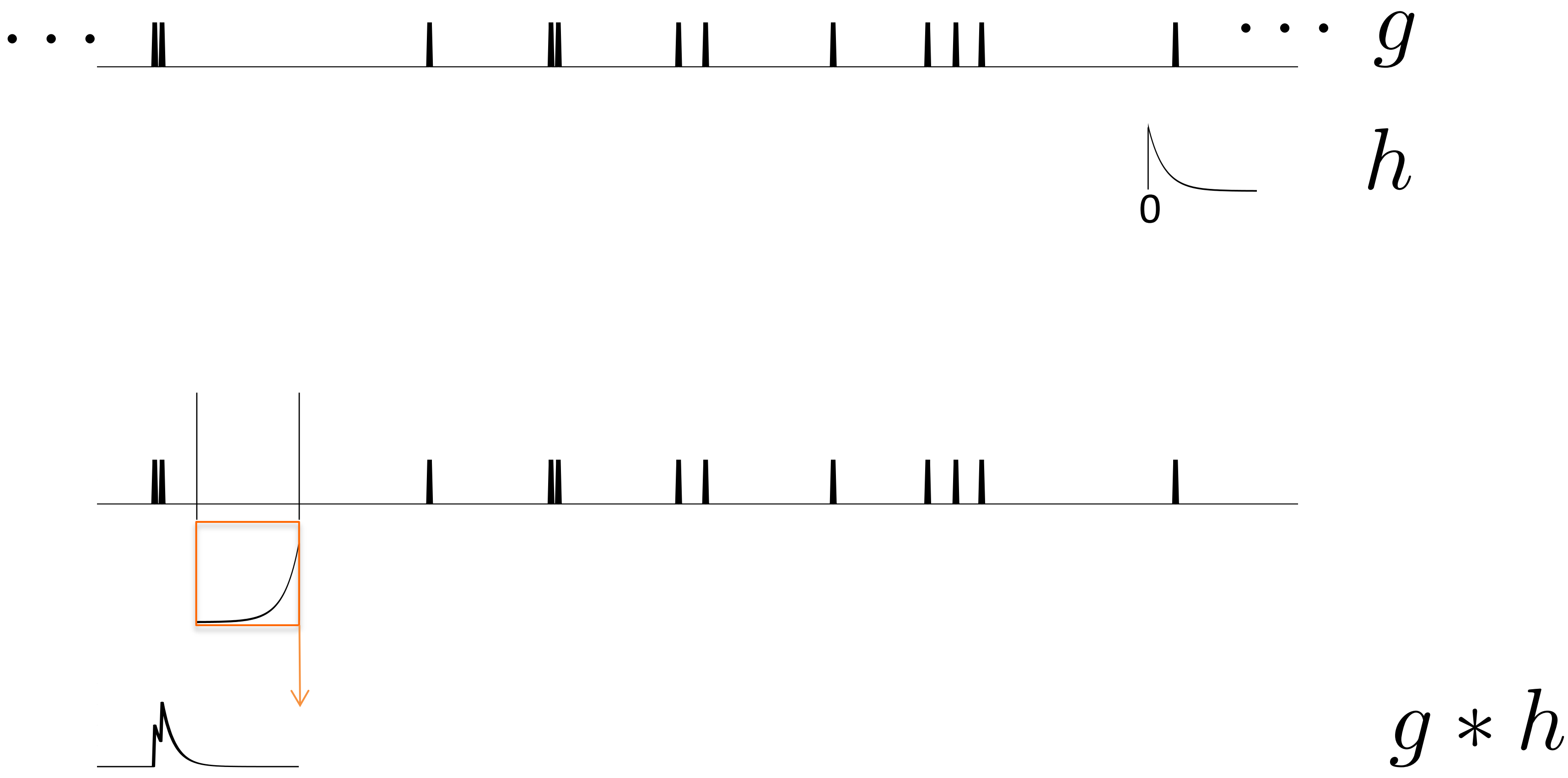
Convolution



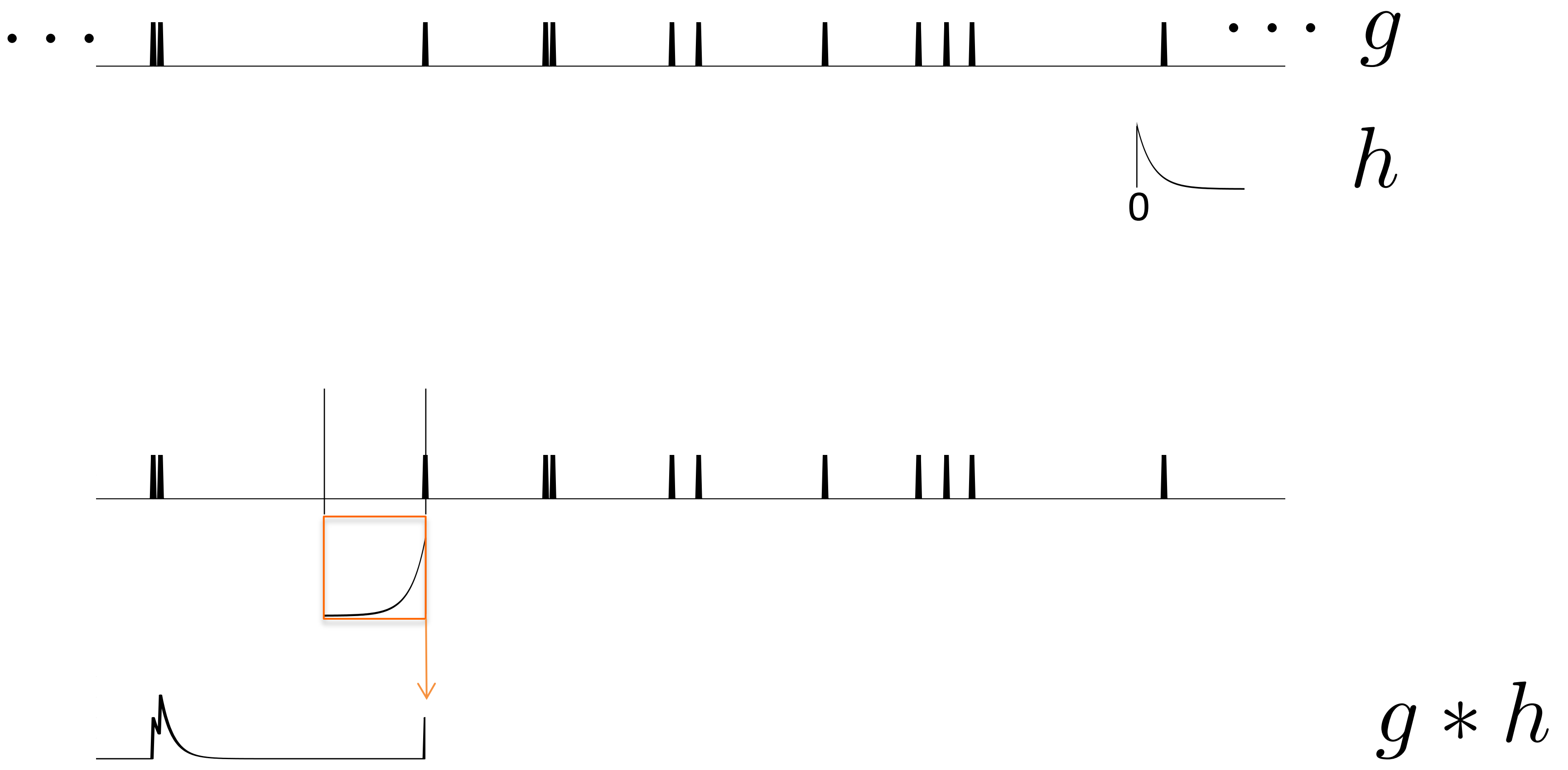
Convolution



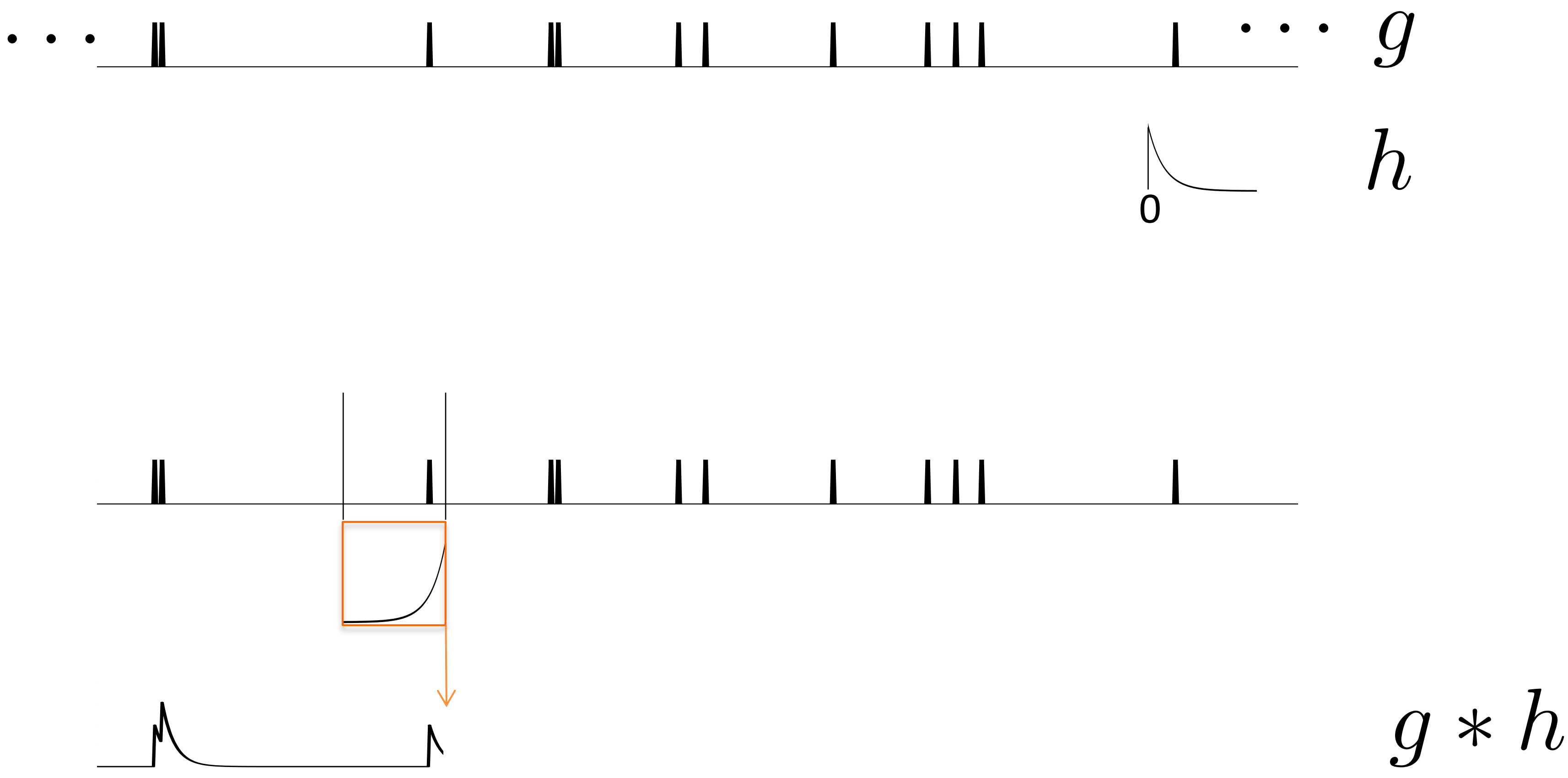
Convolution



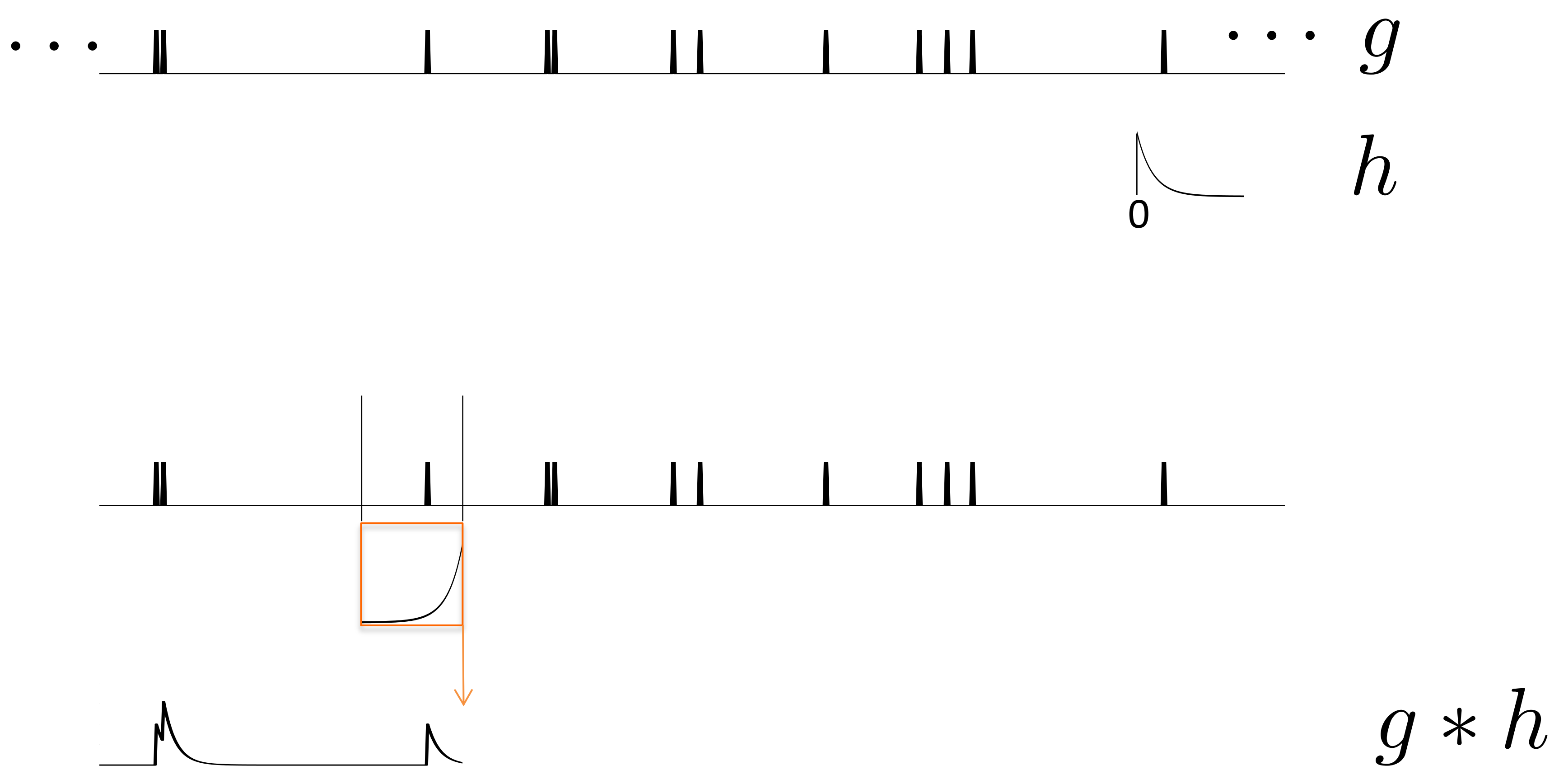
Convolution



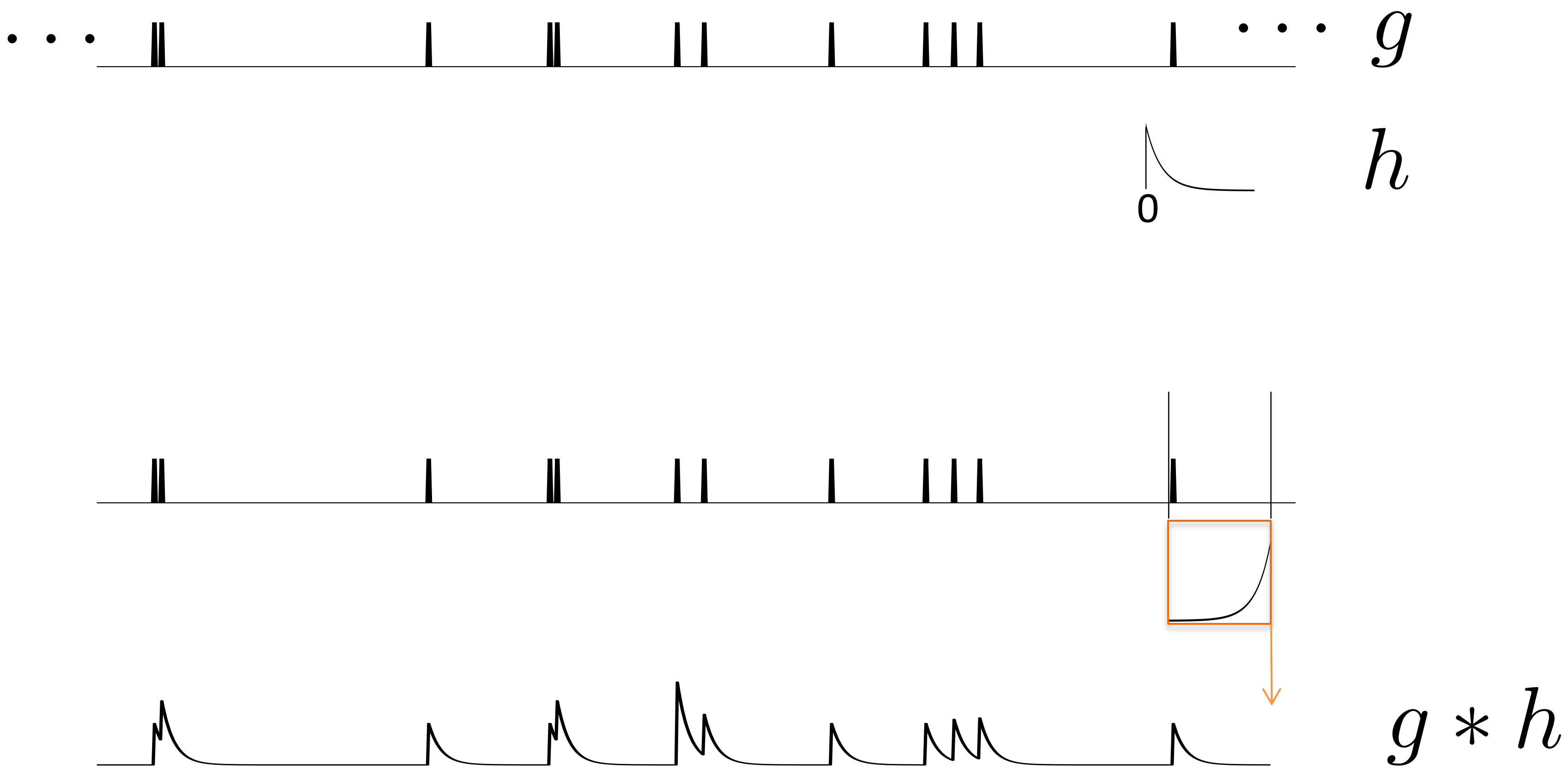
Convolution



Convolution



Convolution



Common convolution kernels

- Boxcar $$h = 1/N \text{ for } N \text{ samples, } 0 \text{ elsewhere}$$

(e.g. rates from spikes)

- Exponential $$h = \frac{1}{\tau} e^{-t/\tau} \text{ for } t > 0, \; 0 \text{ otherwise}$$

(e.g. EPSPs from spikes). Called a linear low-pass filter.

- Gaussian $$h = \frac{1}{\sqrt{2\pi\sigma}} e^{-t^2/2\sigma^2}$$

(e.g. smoothing)

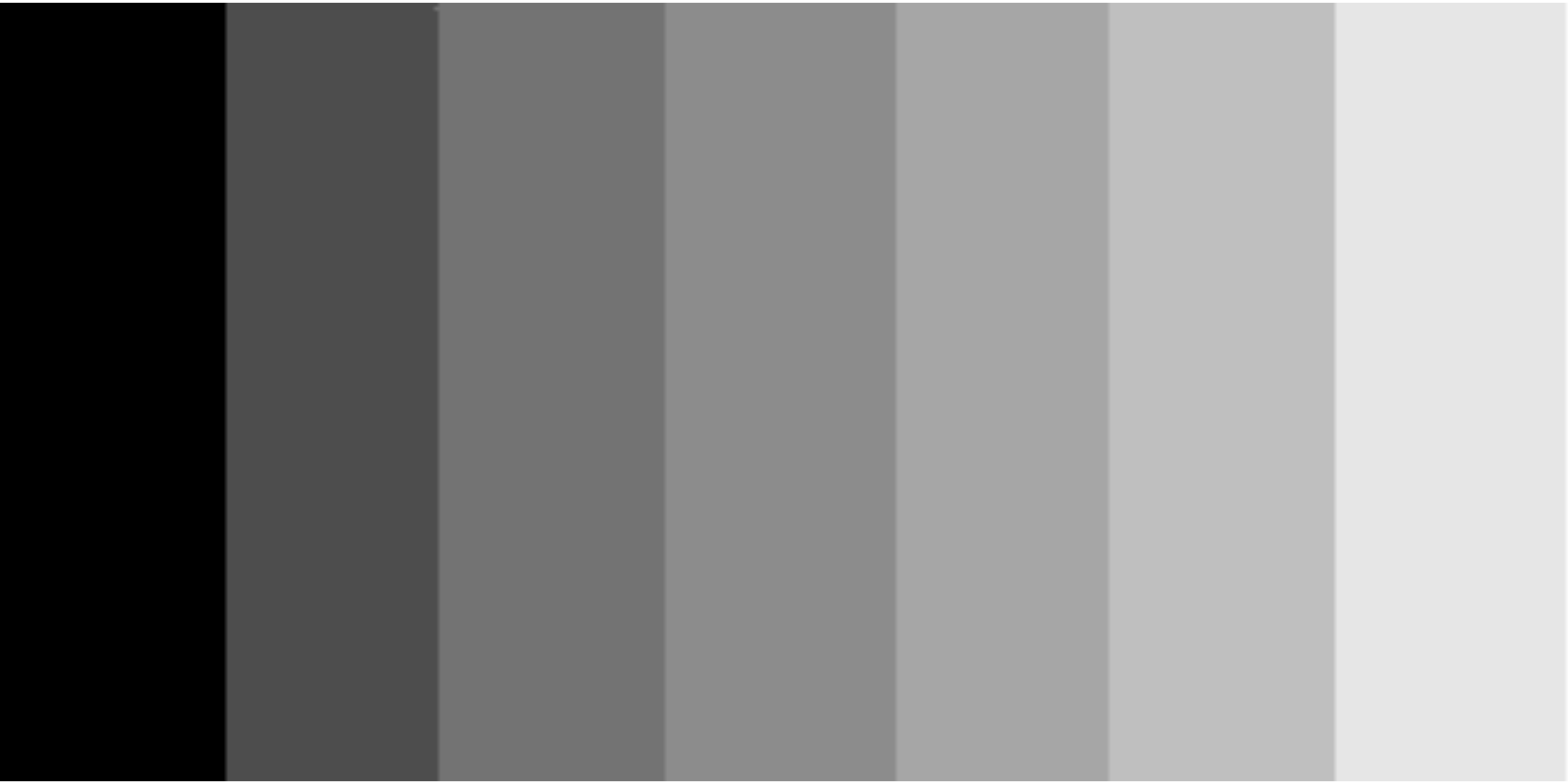
Spikes to rate, smoothing, EPSPs

MATLAB DEMOS

Edge detection, HDR imaging

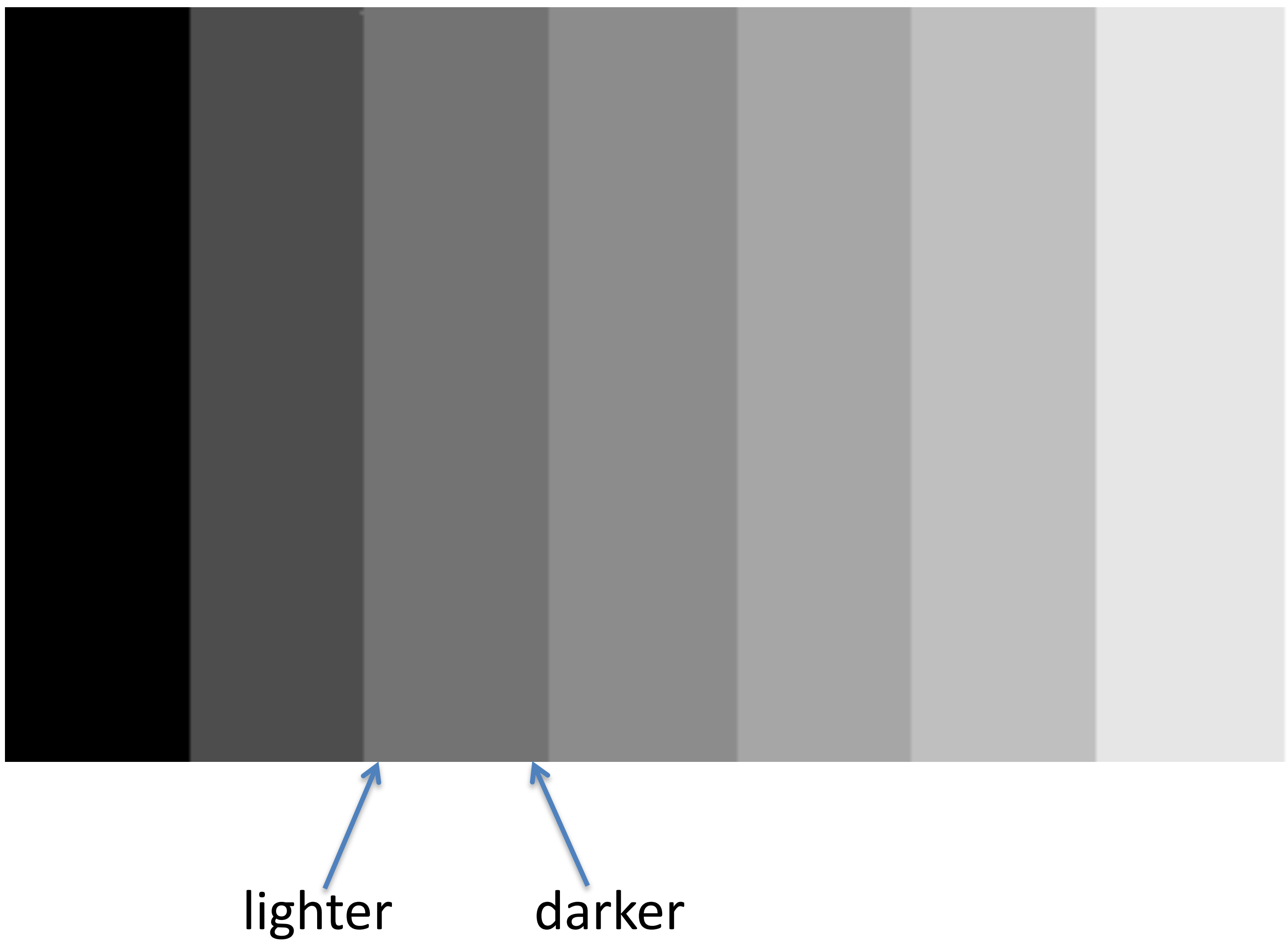
RETINA AS A CONVOLUTIONAL FILTER

Mach bands (Ernst Mach 1860's)



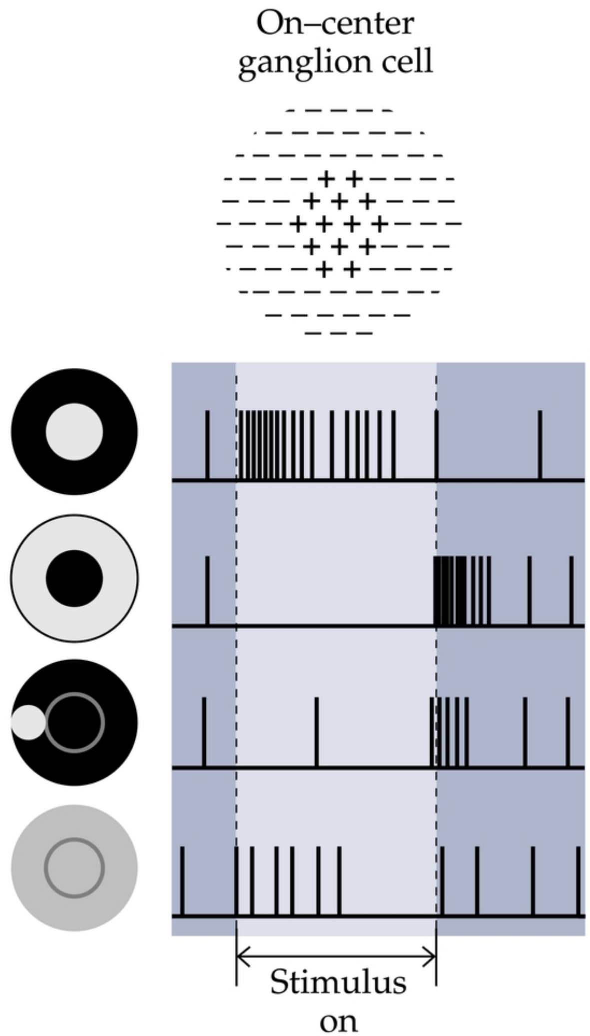
Eight bars of stepped grayscale intensity.
Each bar: constant intensity.

Interesting perceptual effect in Mach bands



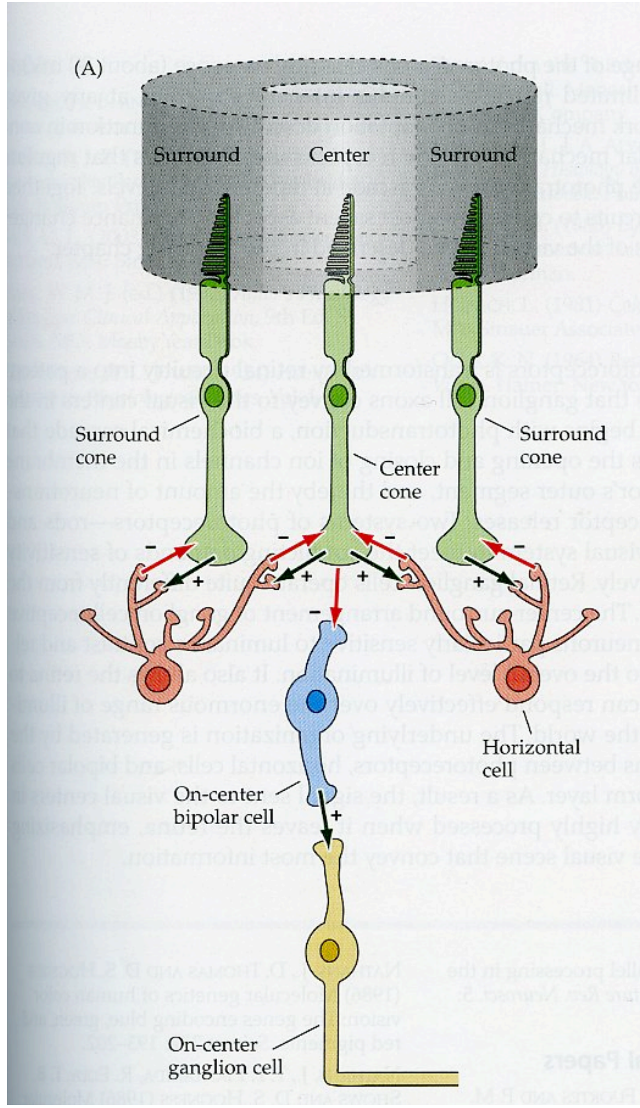
Illumination at a point on the retina is not perceived objectively, but only in reference to its neighbours. Why/how does this happen?

Electrophysiology of a retinal ganglion cell (RGC)



Difference-of-Gaussians or center-surround receptive field.

Anatomy of RGC circuit



Off-center (surround) stimulus has reverse effect of on-center stimulus because of inhibitory horizontal cells.

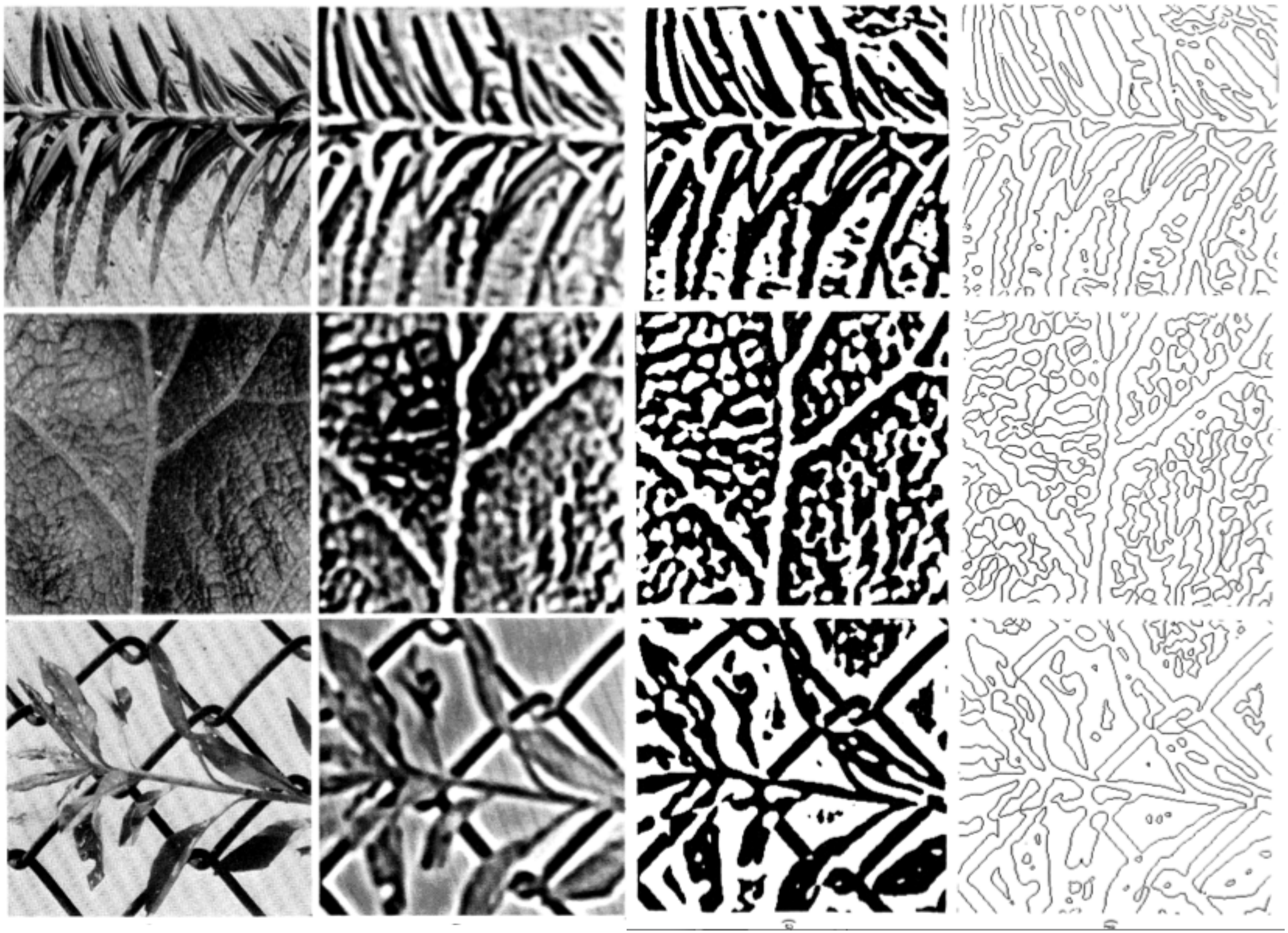
Reproducing the Mach band illusion

RETINAL KERNEL SENSES CHANGES IN ILLUMINATION (MATLAB)

Ganglion cells code **contrast:** difference in brightness between center and surround

Retinal filter performs edge detection

196 D. Marr and E. Hildreth



Demo

RETINAL KERNEL AS EDGE DETECTOR (MATLAB)

How edge detection works: theory

Smoothing filter: $$H_\sigma(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-x^2/2\sigma^2}$$ Gaussian

Retinal filter model: $$H_{retinal}(x) = \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-x^2/2\sigma_1^2} - \alpha \frac{1}{\sqrt{2\pi\sigma_2^2}} e^{-x^2/2\sigma_2^2}$$
$$= H_{\sigma_1}(x) - H_{\sigma_2}(x)$$
$\sigma_1 < \sigma_2$

difference-of-Gaussians

How to interpret Retinal/difference-of-Gaussians filter?

How edge detection works: theory

$$-\frac{d^2}{dx^2}H_\sigma(x) = -\frac{d^2}{dx^2}\frac{1}{\sqrt{2\pi\sigma^2}}e^{-x^2/2\sigma^2}$$

$$= \frac{1}{\sigma^2}\left(H_\sigma(x) - \frac{x^2}{\sigma^2}H_\sigma(x)\right)$$

$$\approx H_{\sigma_1}(x) - H_{\sigma_2}(x)$$

difference-of-Gaussians with some $\sigma_2 < \sigma_1$

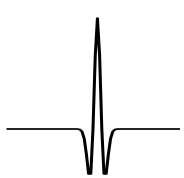
D. Marr, E. Hildreth (1980) "Theory of edge detection."
Proc. R. Soc. Lond. B, 207:187-217.

How edge detection works: theory

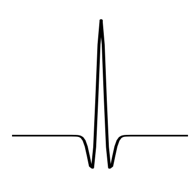
$$-\frac{d^2}{dx^2}H_\sigma(x) = -\frac{d^2}{dx^2}\frac{1}{\sqrt{2\pi\sigma^2}}e^{-x^2/2\sigma^2}$$

$$= \frac{1}{\sigma^2}\left(H_\sigma(x) - \frac{x^2}{\sigma^2}H_\sigma(x)\right)$$

$$\approx H_{\sigma_1}(x) - H_{\sigma_2}(x)$$



$\approx$



$\sigma_2 = \frac{3}{2}\sigma_1$

How edge detection works: theory

$$H_{retinal} \approx (H_{\sigma_1}(x) - H_{\sigma_2}(x)) \approx -\frac{d^2}{dx^2} H_\sigma(x)$$

2nd derivative　　smoothing

Retinal filter (difference-of-Gaussians) is like smoothing filter followed by a 2nd derivative filter:

$$H_{retinal} \approx H_{2nd\ diff} * H_{smooth}$$

High dynamic-range imaging

Retinex-based adaptive filter:
global compression, local processing

Comparison: convolution, cross-correlation, autocorrelation

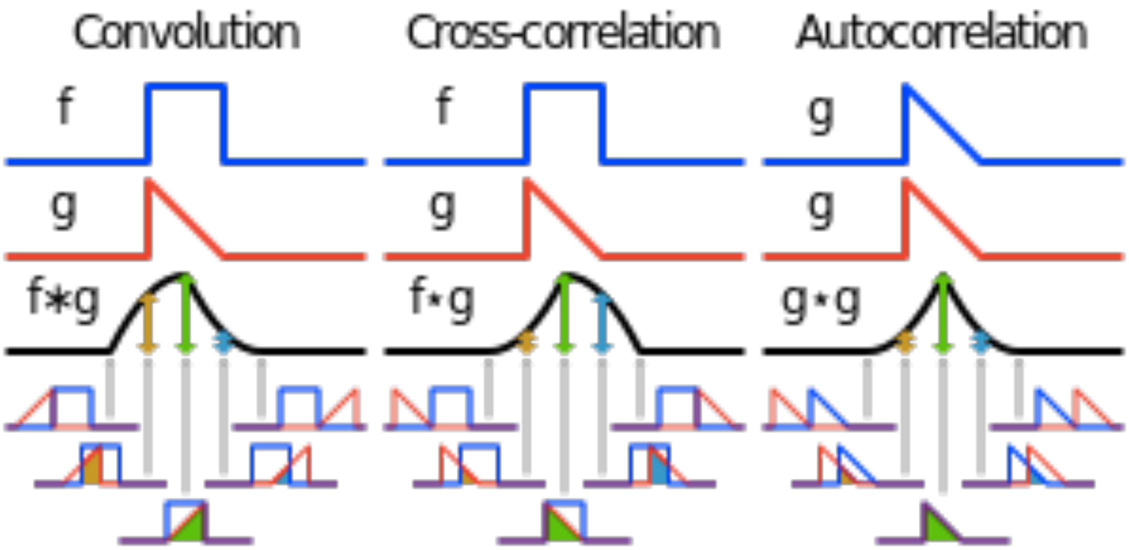


Image: wikimedia commons
https://en.wikipedia.org/wiki/Convolution

Summary

- Convolution: a kernel (short) acts on a signal (long), to produce a locally reweighted version of the signal.
- Useful in engineering sense: smooth signals, extract rates from spikes, template matching, other processing.
- Operations of retina on visual stimulus may be interpreted as convolution.
- Retinal difference-of-Gaussians convolution: edge enhancement, edge detection, contrast normalization.