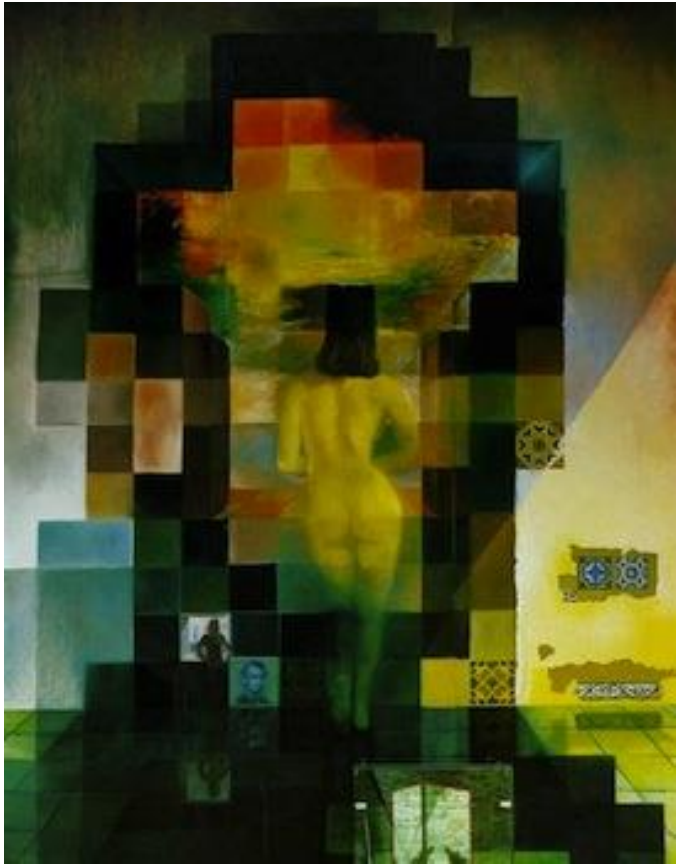
Image Pyramid

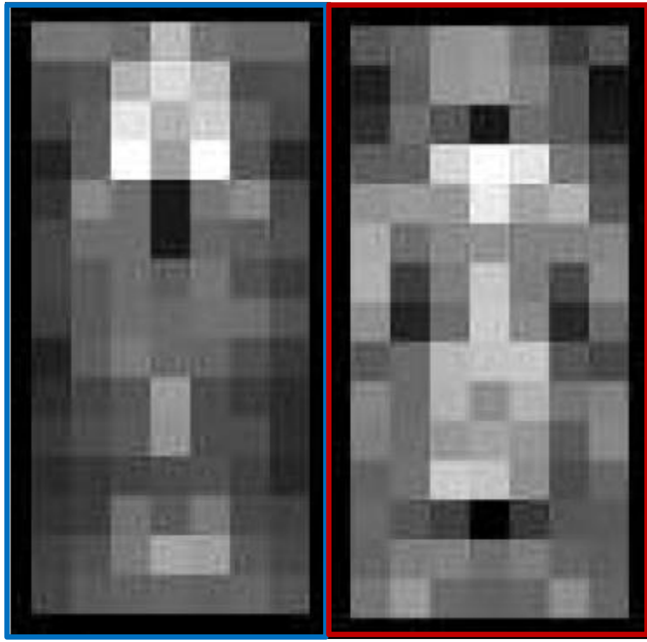
Hyun Soo Park

Salvador Dali, Abraham Lincoln

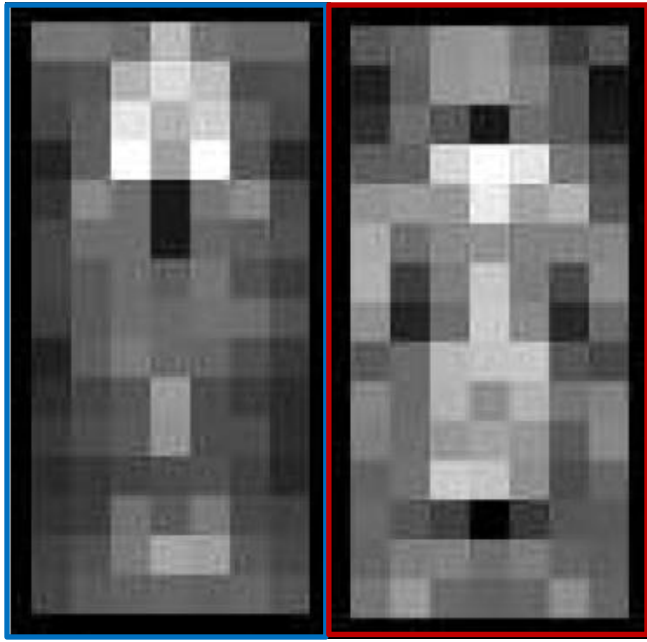


Salvador Dali, Abraham Lincoln

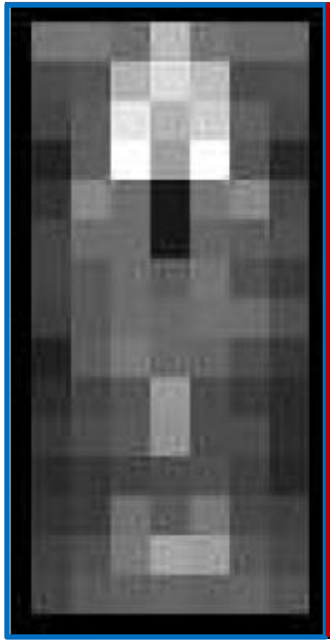
Recall: Object Recognition with HOG

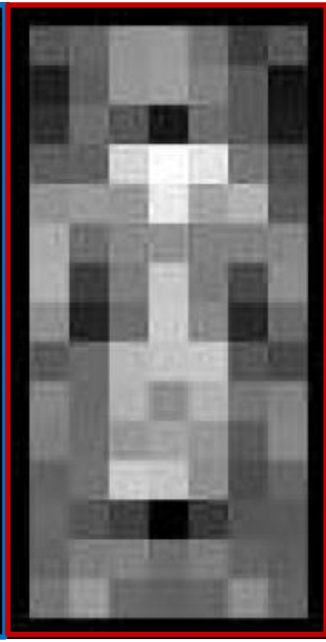


RECALL: OBJECT RECOGNITION WITH HOG

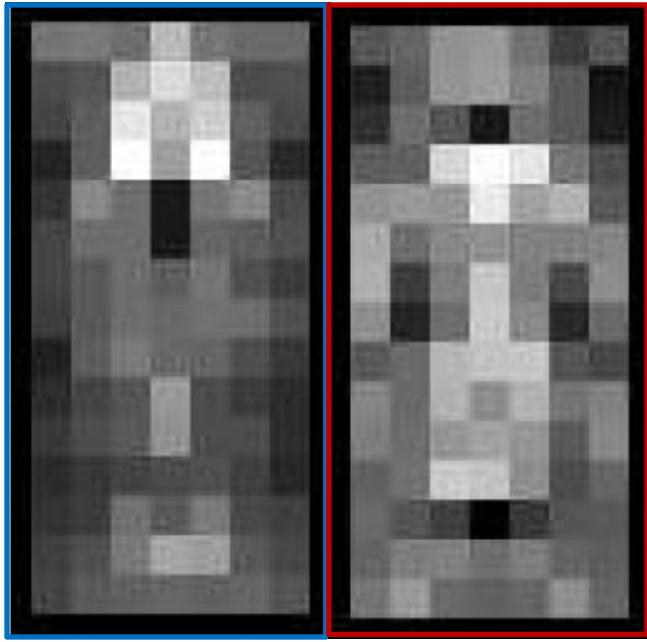


RECALL: OBJECT RECOGNITION WITH HOG

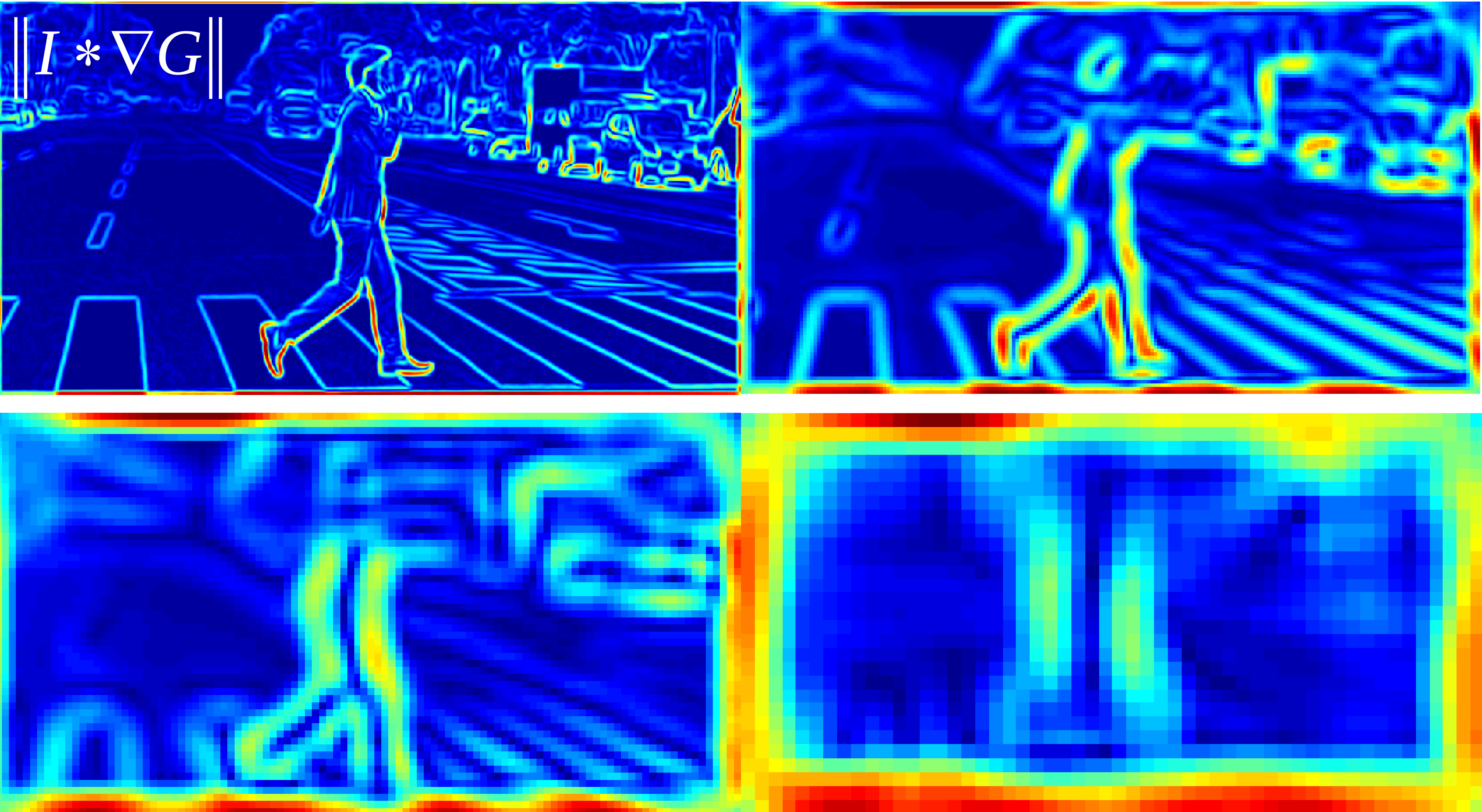




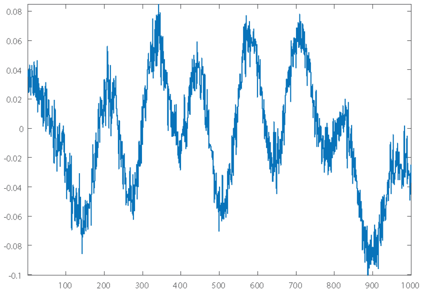
Recall: Object Recognition with HOG





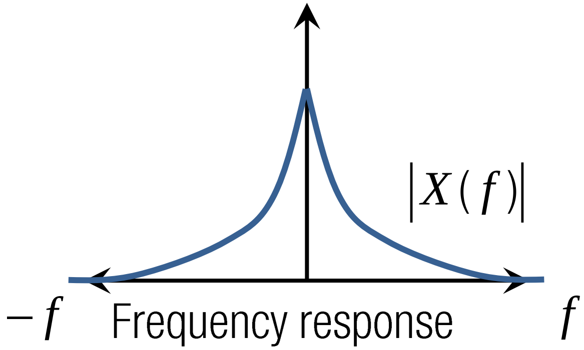


$\|I * \nabla G\|$

Fourier Transform



FT →

Inverse FT ←

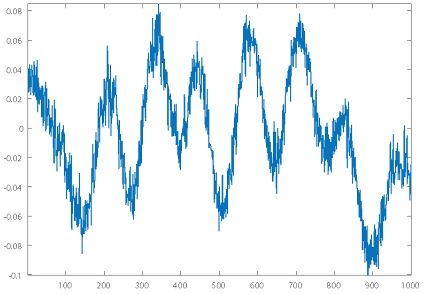


Time signal

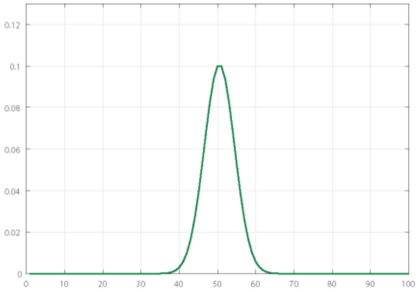
$x(t)$

$X(f)$

Fourier Transform

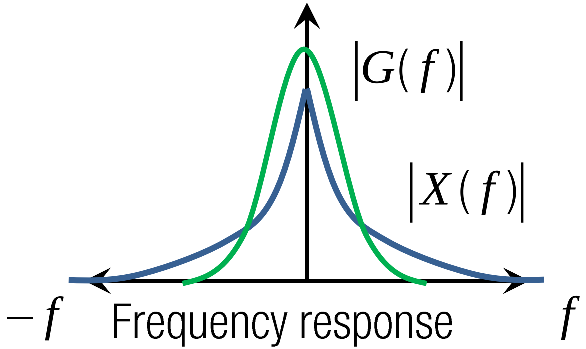


*



FT →

Inverse FT ←



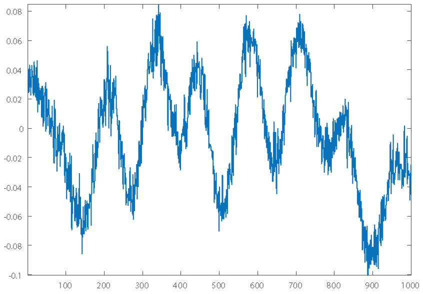
Time signal

Guassian filter

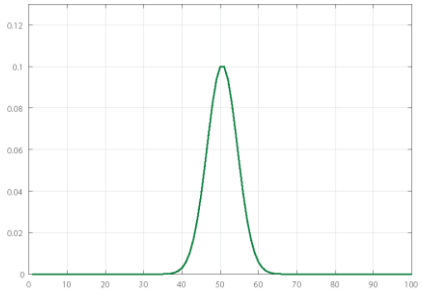
$x(t)$ * $g(t)$

$X(f)\,G(f)$

FOURIER TRANSFORM

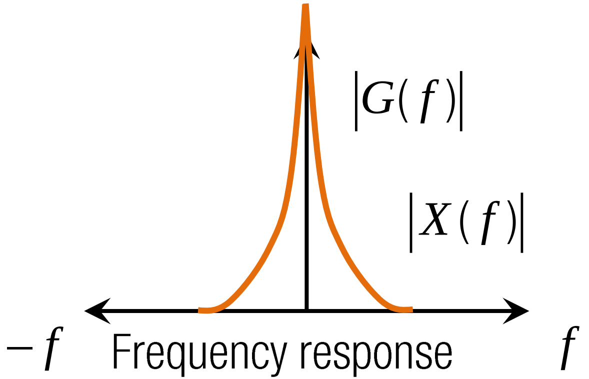


*



FT →

← Inverse FT



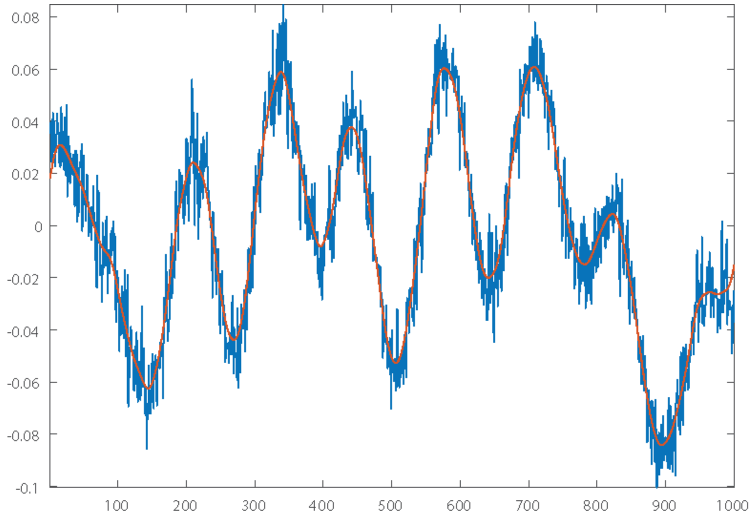
Time signal

Guassian filter

$x(t)$ $*$ $g(t)$

$X(f)\,G(f)$

Fourier Transform



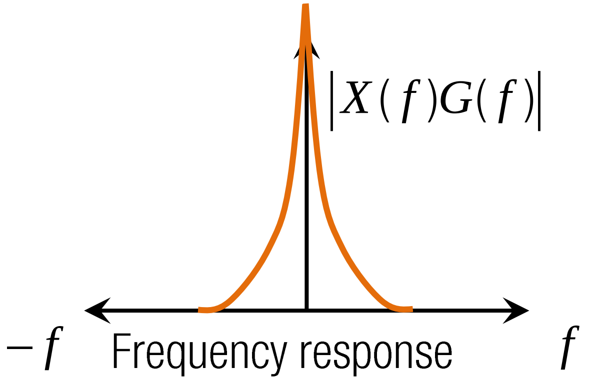
Time signal

Guassian filter

$x(t)$ $*$ $g(t)$

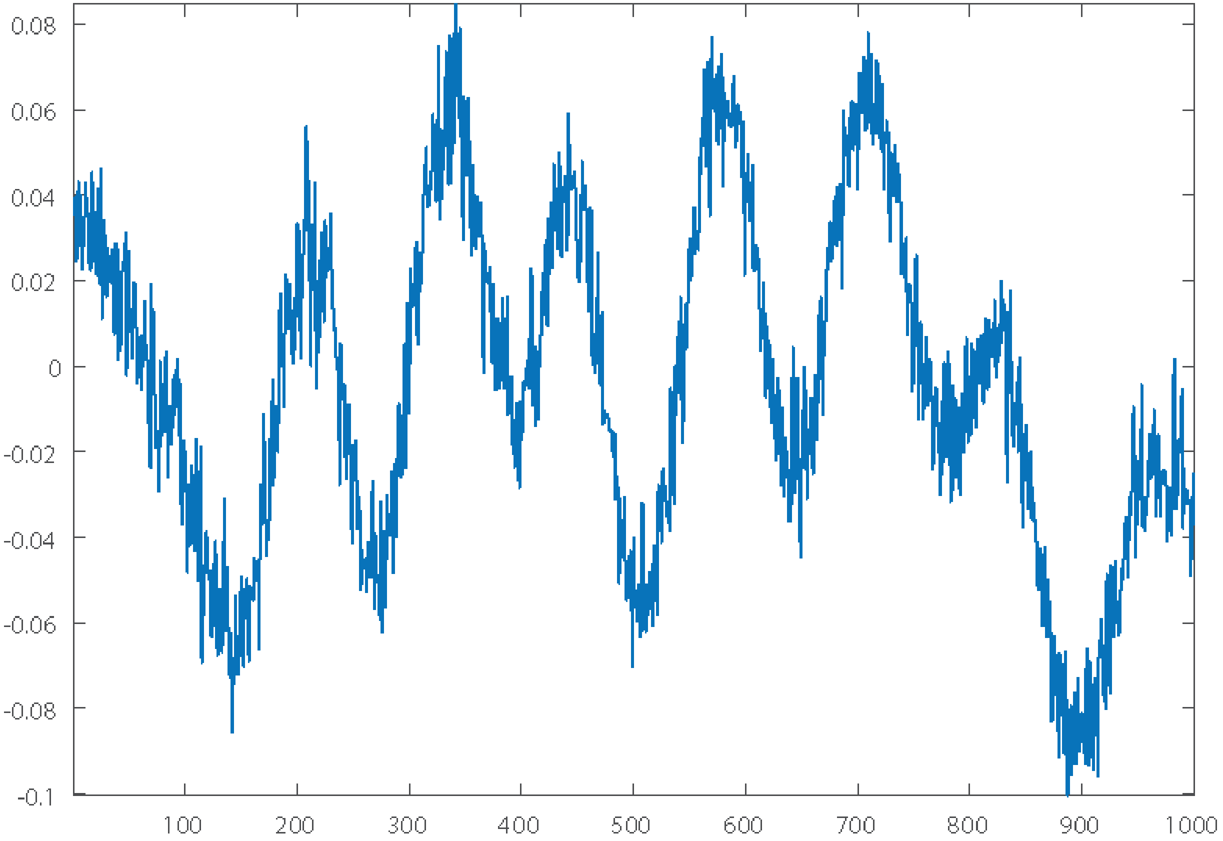
FT →

← Inverse FT

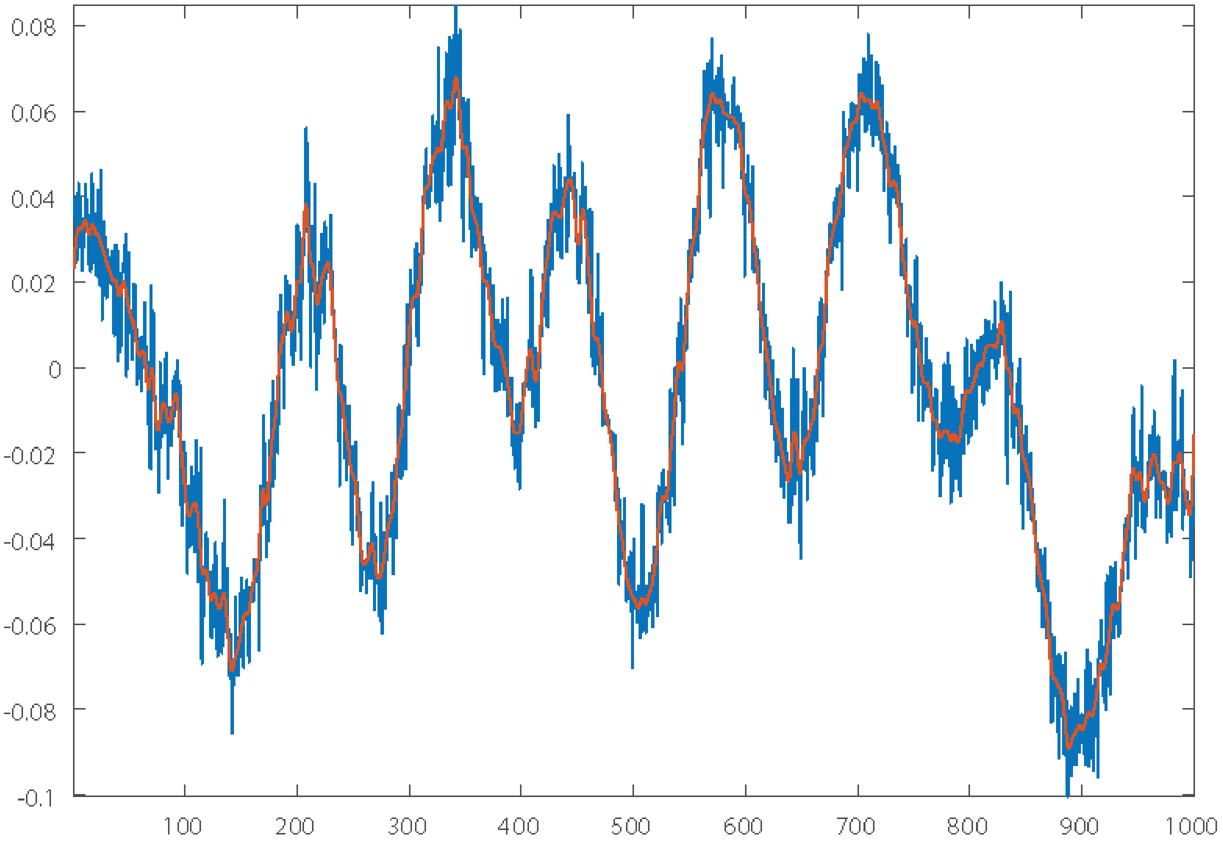


$X(f)\,G(f)$

Gaussian Filtering ~ Low-pass Filtering

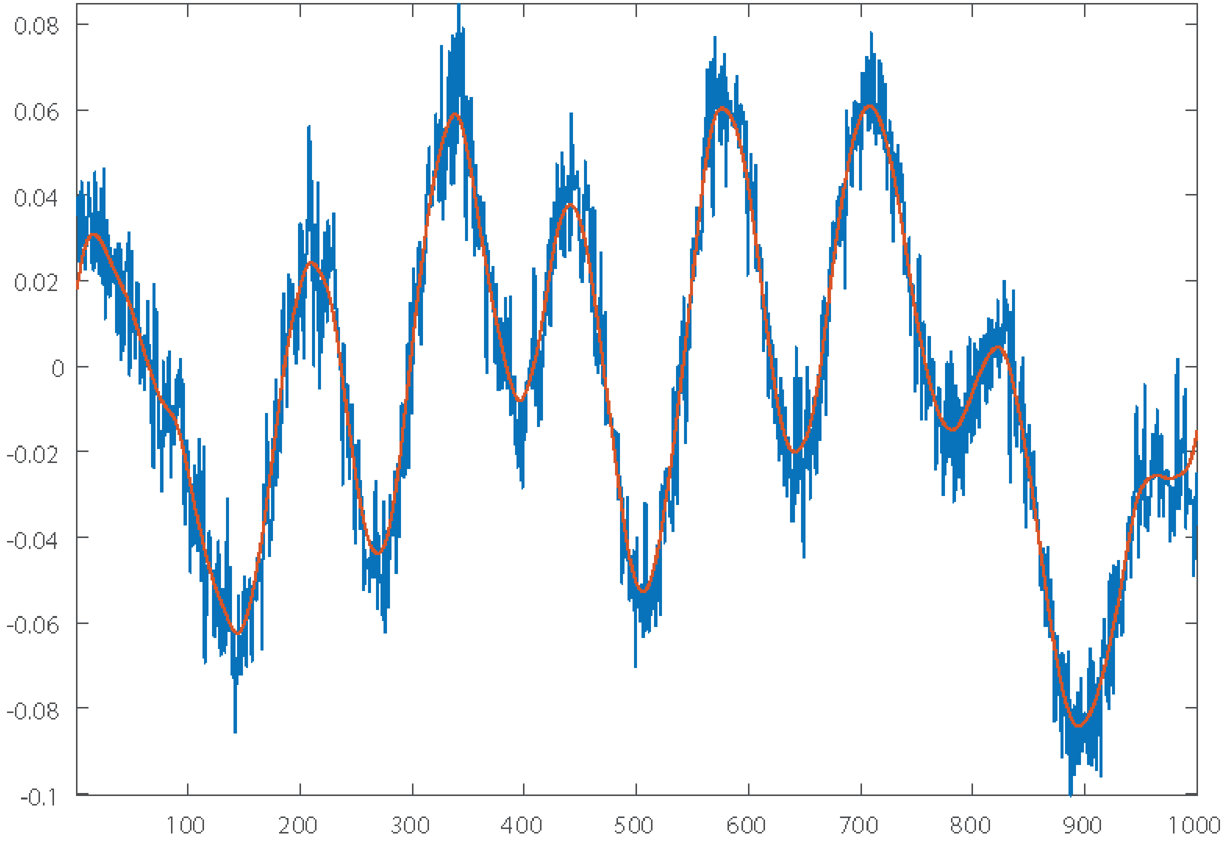


Gaussian Filtering ~ Low-pass Filtering



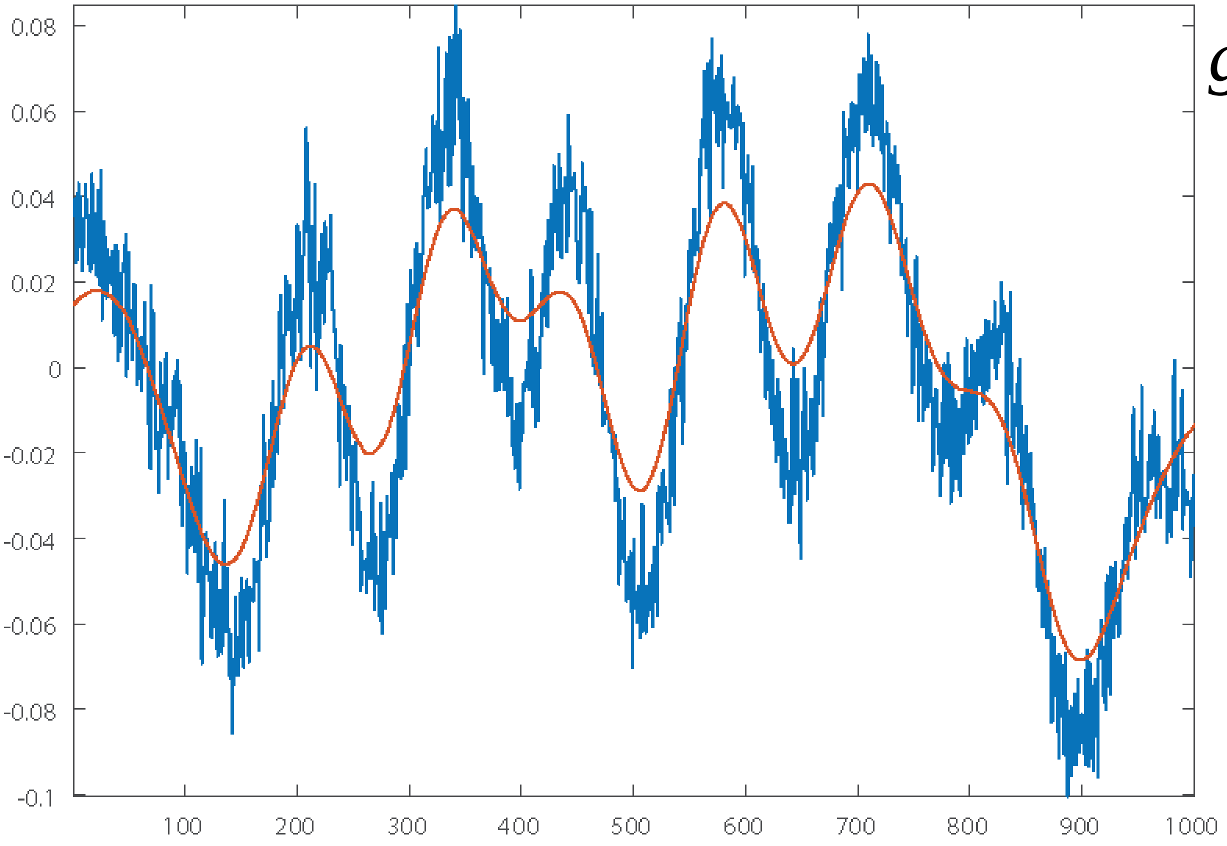
$$g(\sigma = 2)$$

Gaussian Filtering ~ Low-pass Filtering



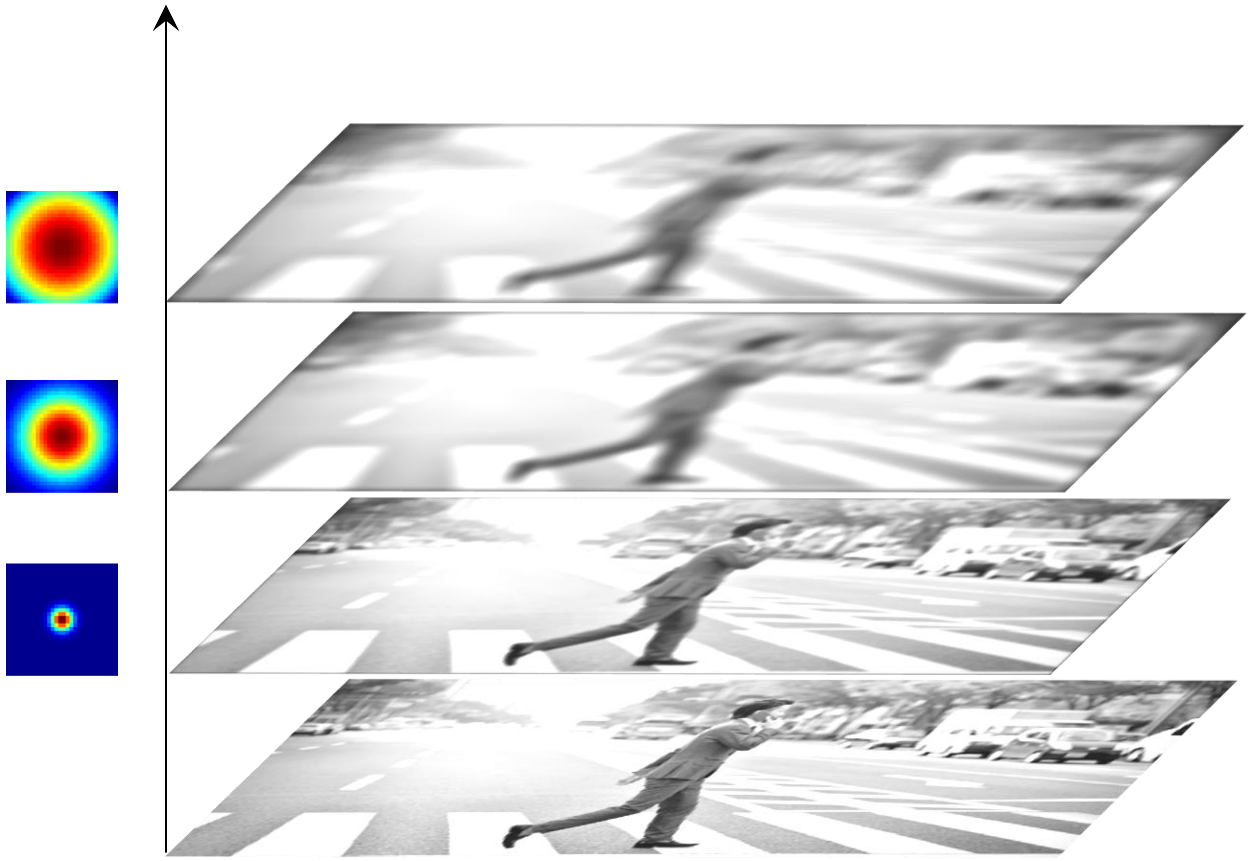
$g(\sigma = 8)$

GAUSSIAN FILTERING ~ LOW-PASS FILTERING



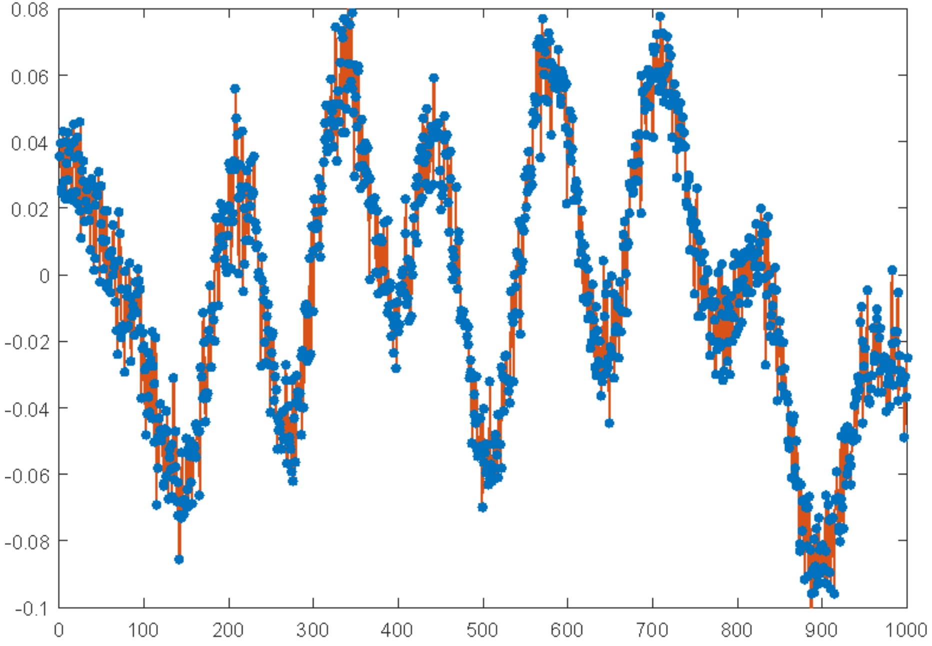
$g(\sigma = 32)$

Gaussian Filtering ~ Low-pass Filtering ~ Image Blurring



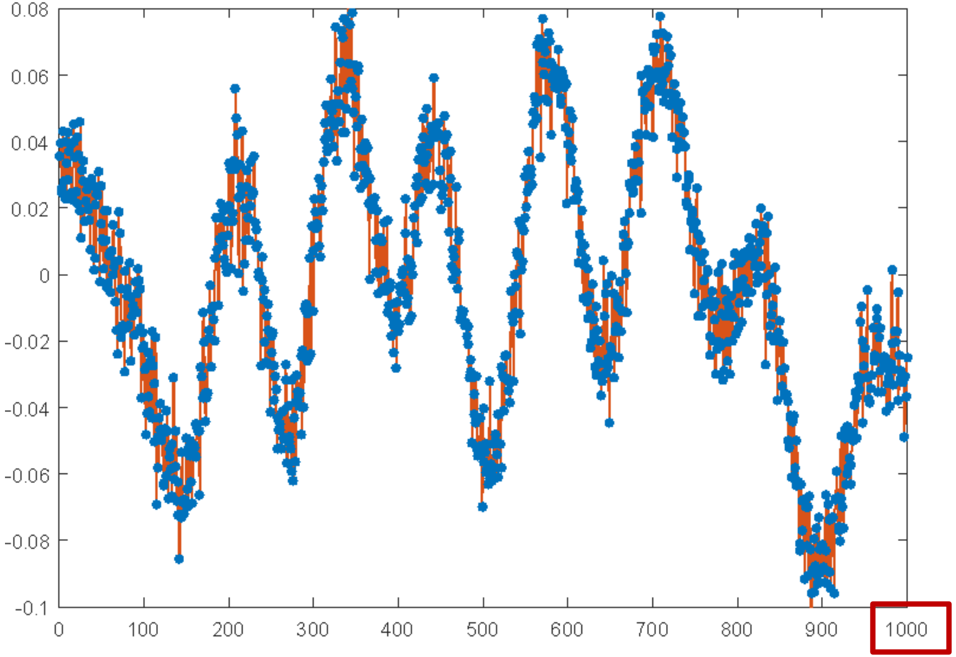
MULTI-DIMENSIONAL IMAGE REPRESENTATION

SUBSAMPLING



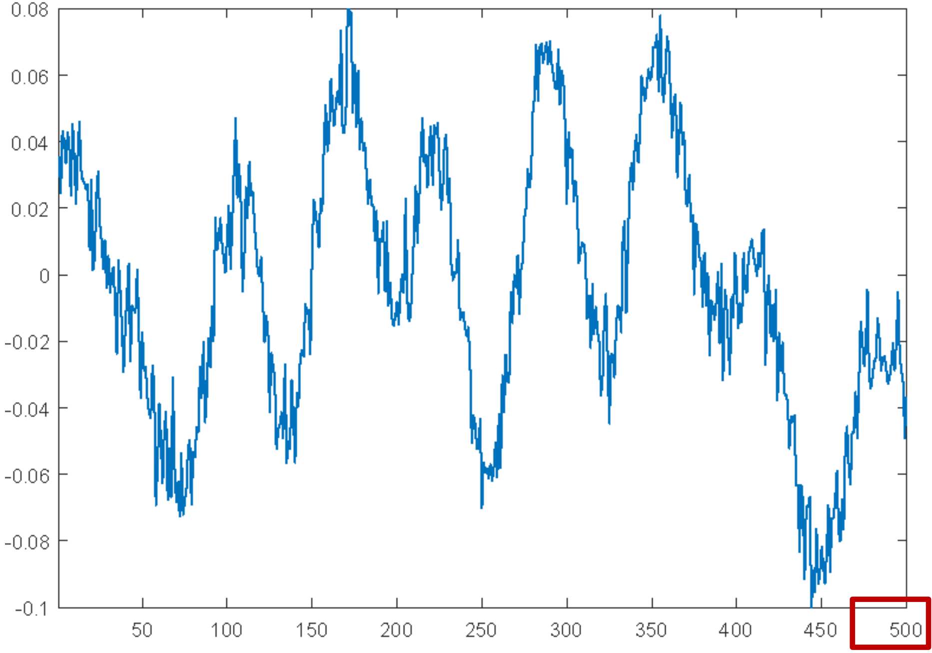
$x(t)$

SUBSAMPLING



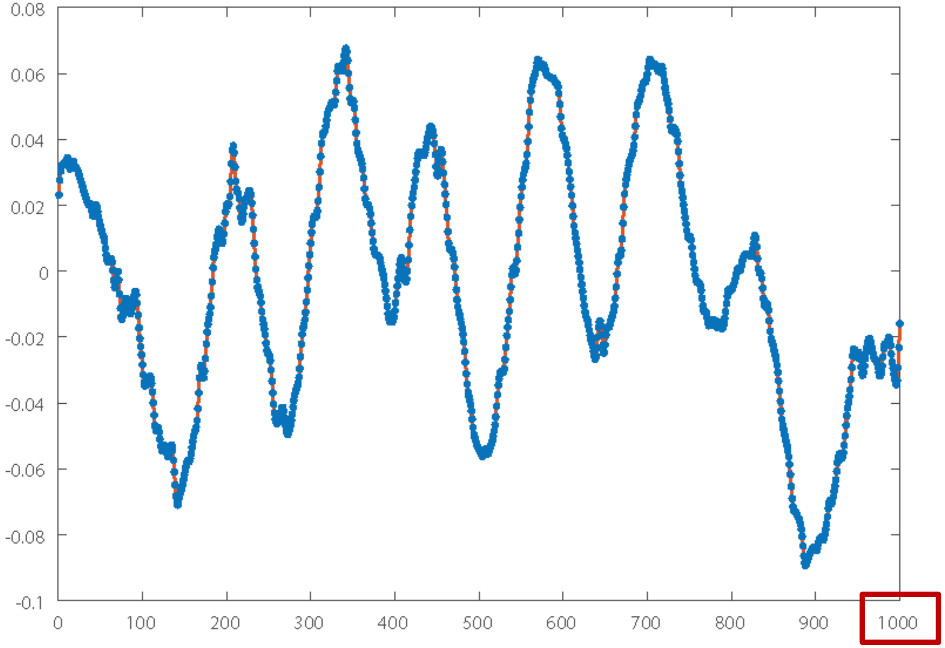
$x(t)$

Subsampling



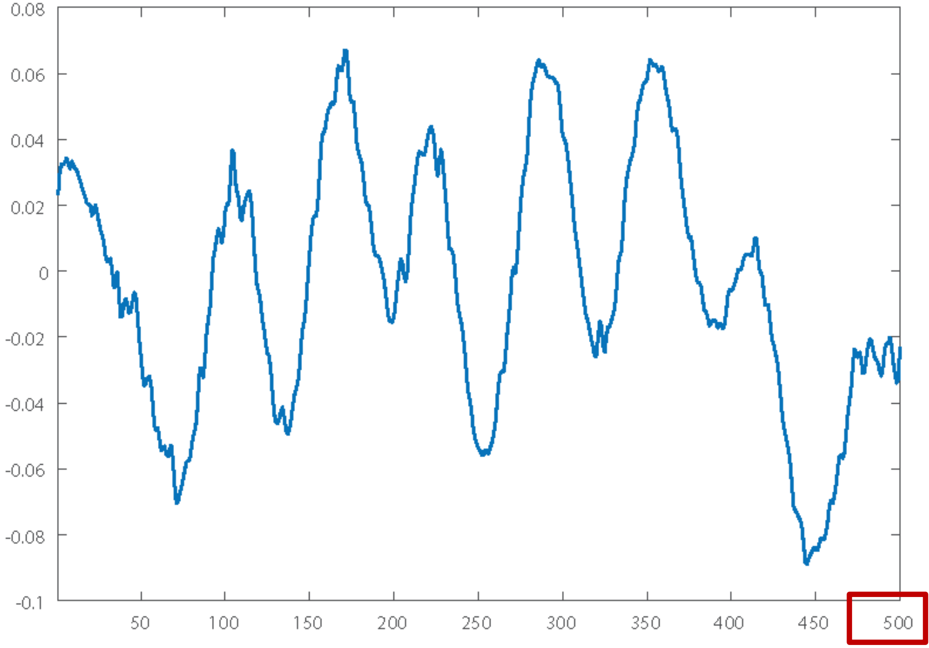
$\downarrow x(t)$

SUBSAMPLING WITH G. FILTERING



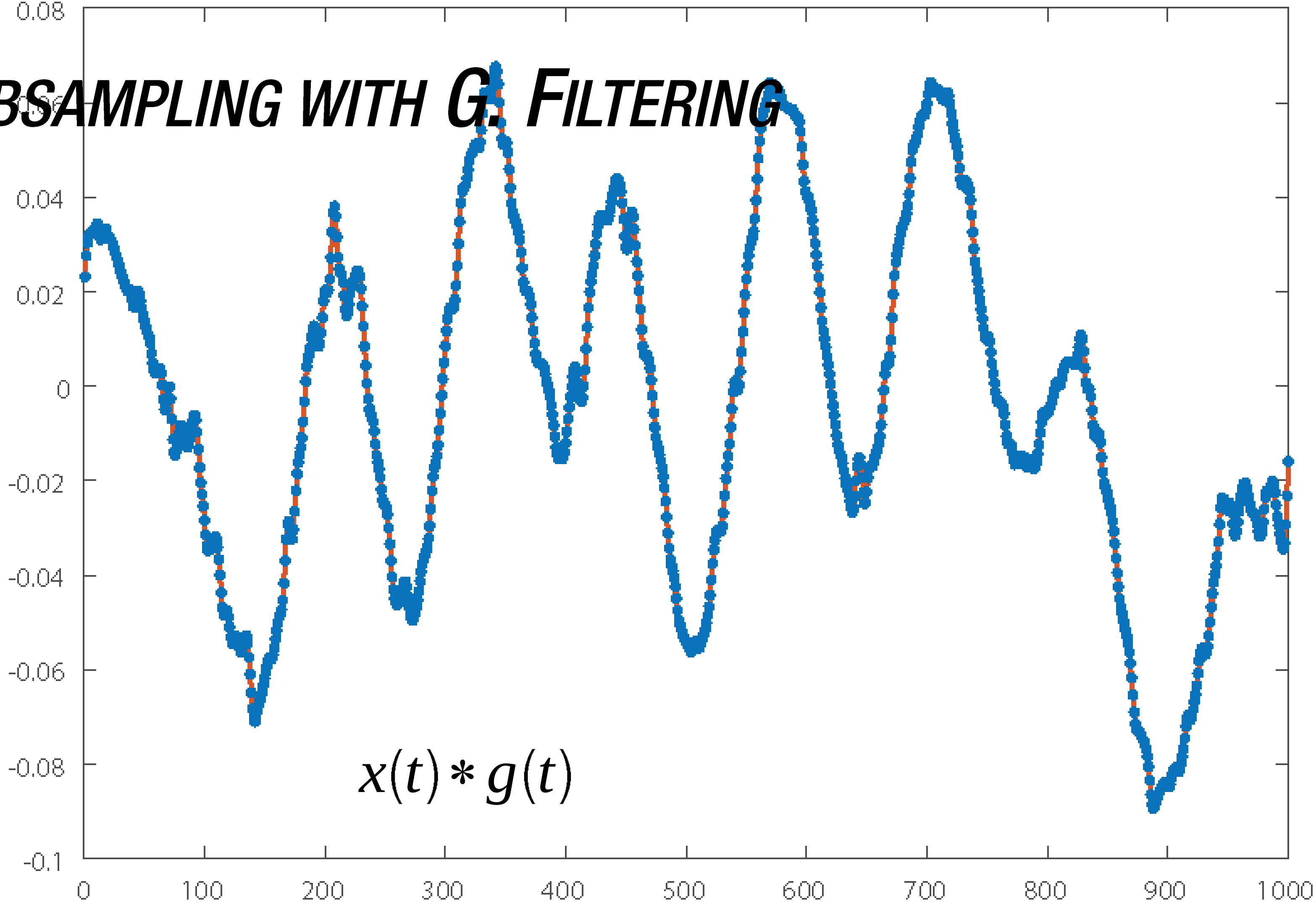
$x(t) * g(t)$

$\longrightarrow$ Subsampling

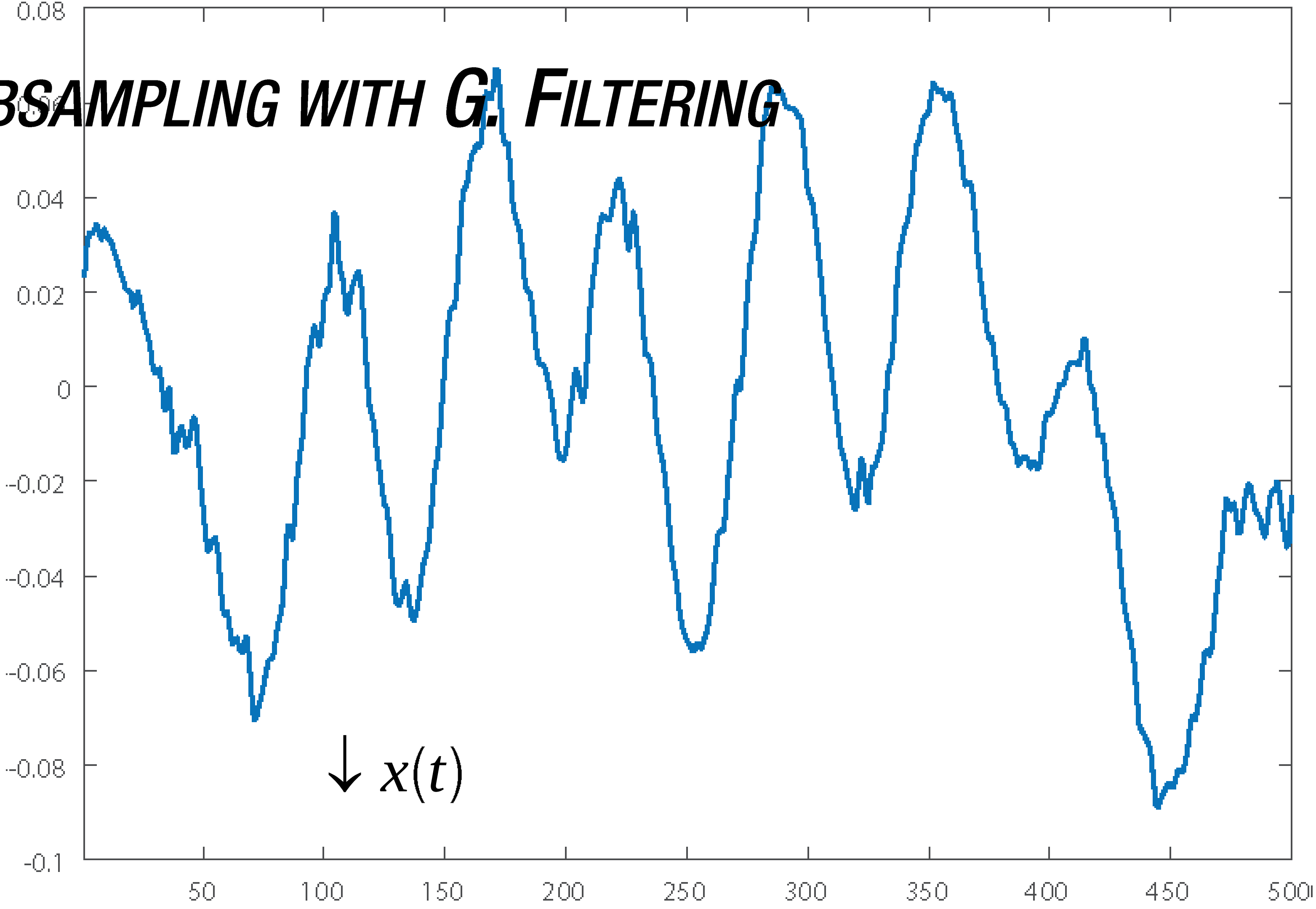


$\downarrow x(t)$

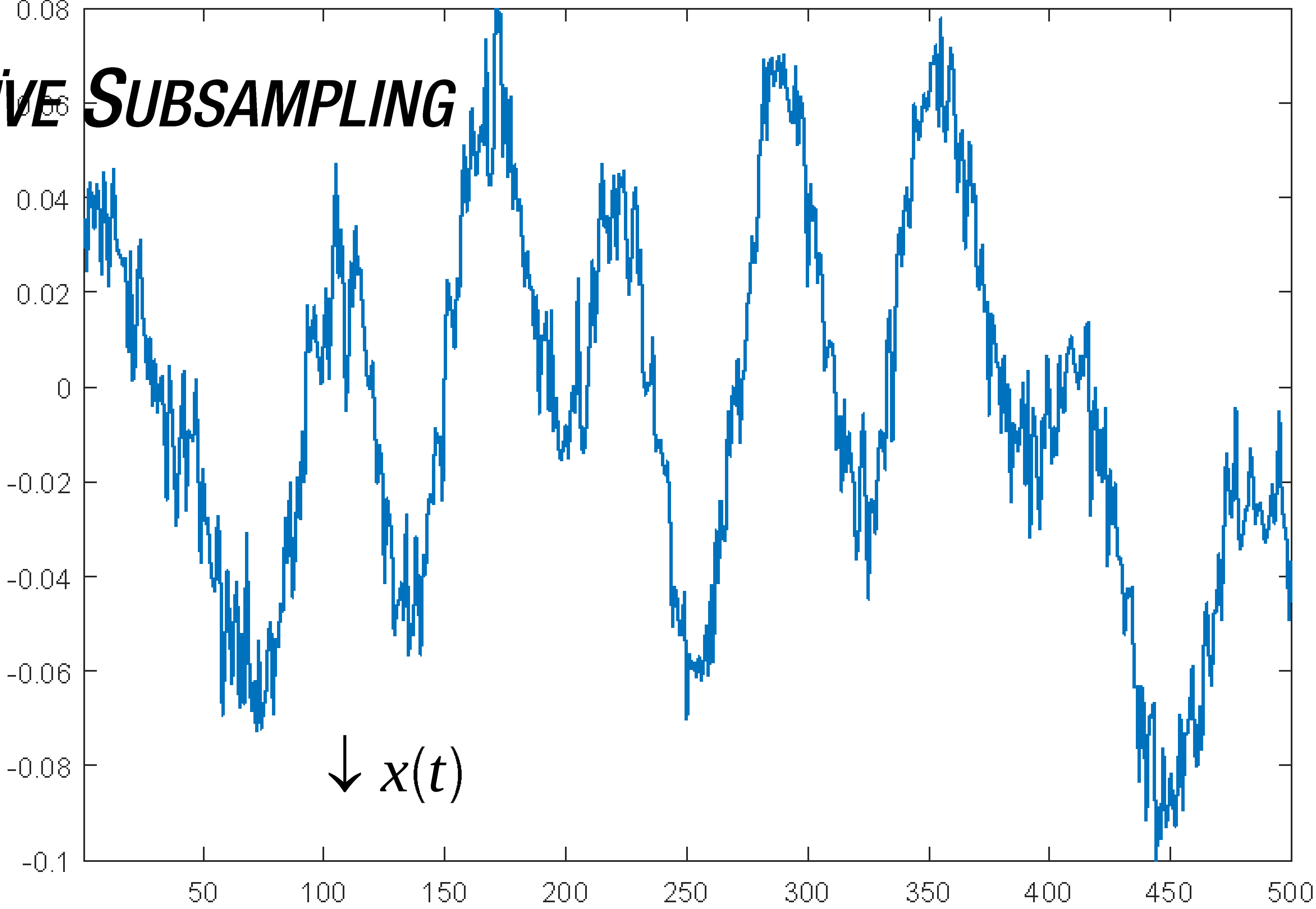
SUBSAMPLING WITH G. FILTERING



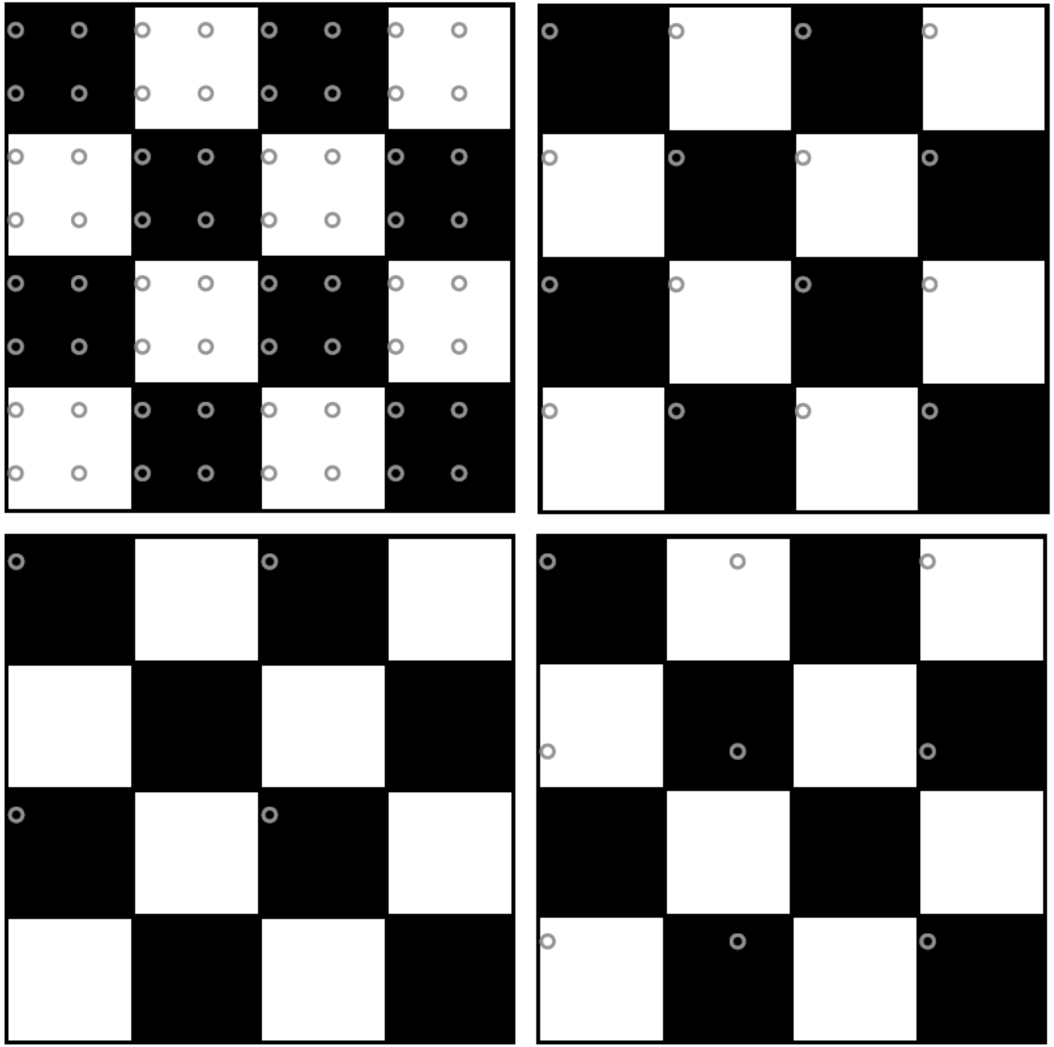
SUBSAMPLING WITH G. FILTERING



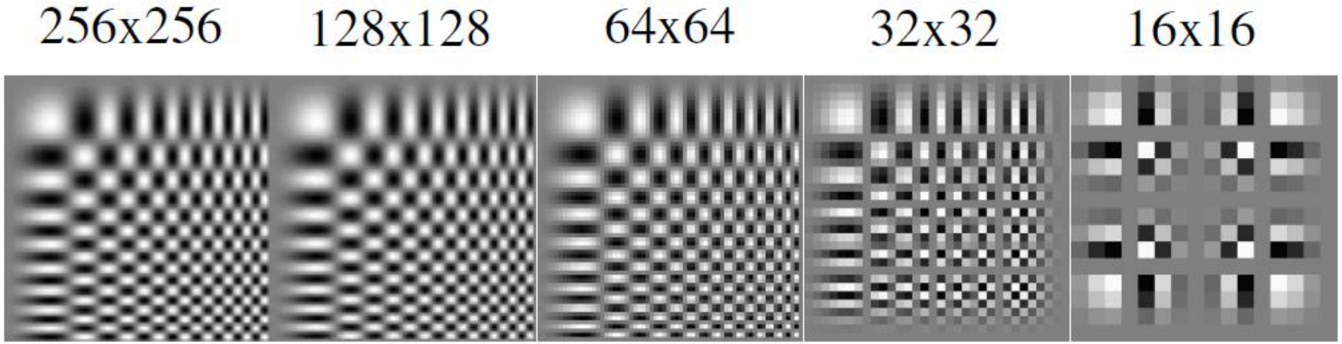
Naïve Subsampling



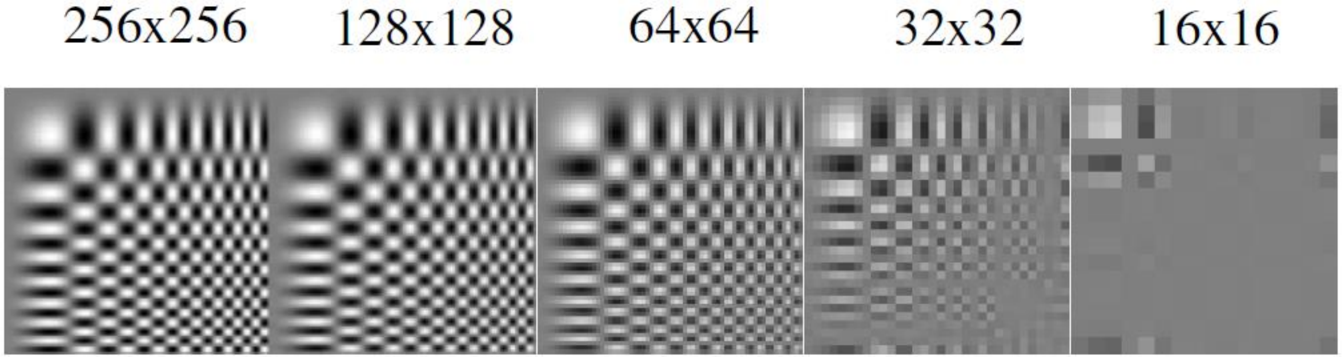
ALIASING



ALIASING

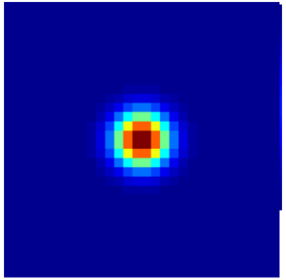


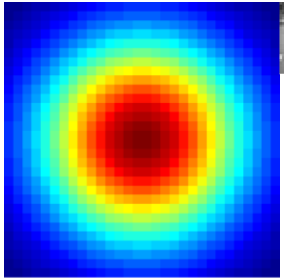
Naïve subsampling

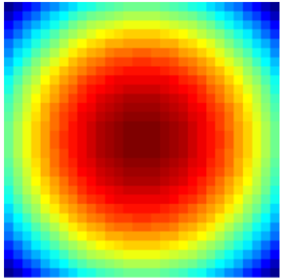


Smoothing and subsampling: eliminating aliasing effects.





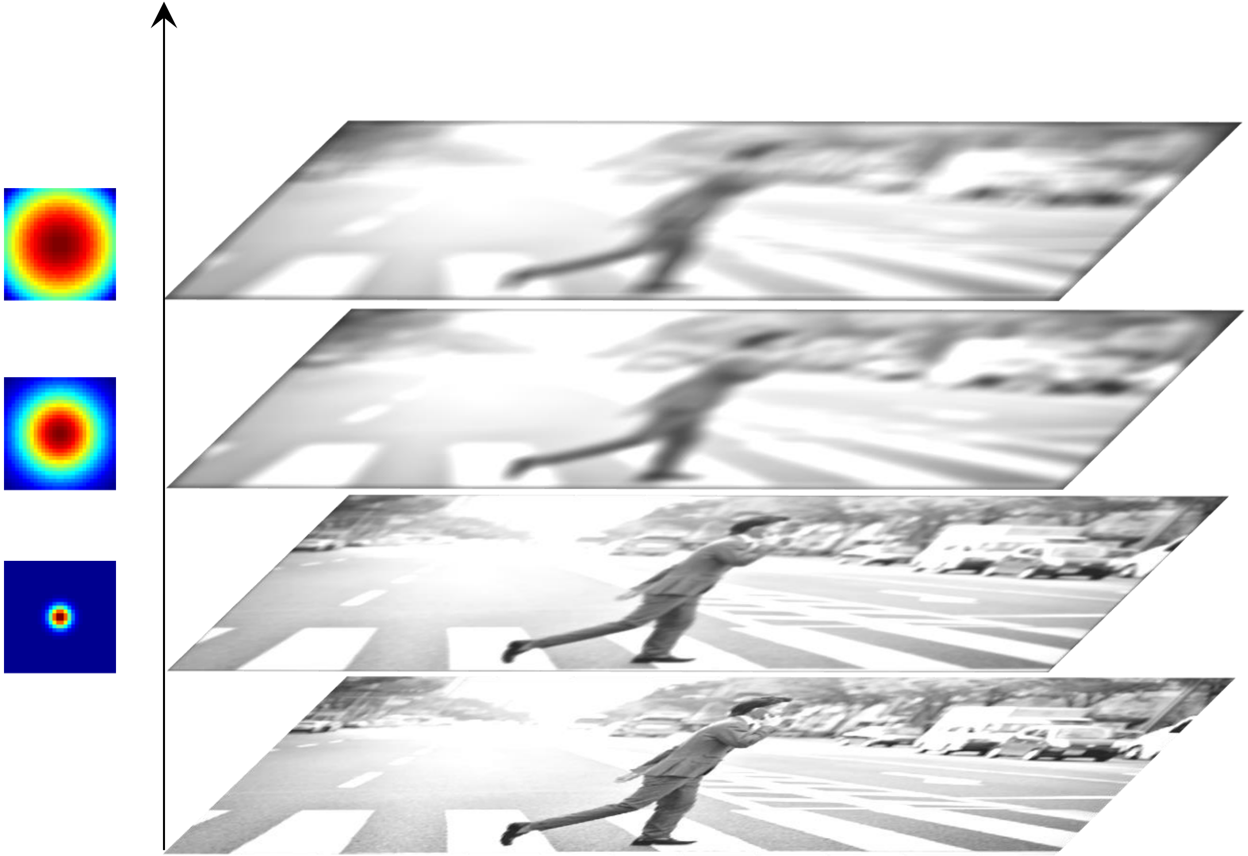




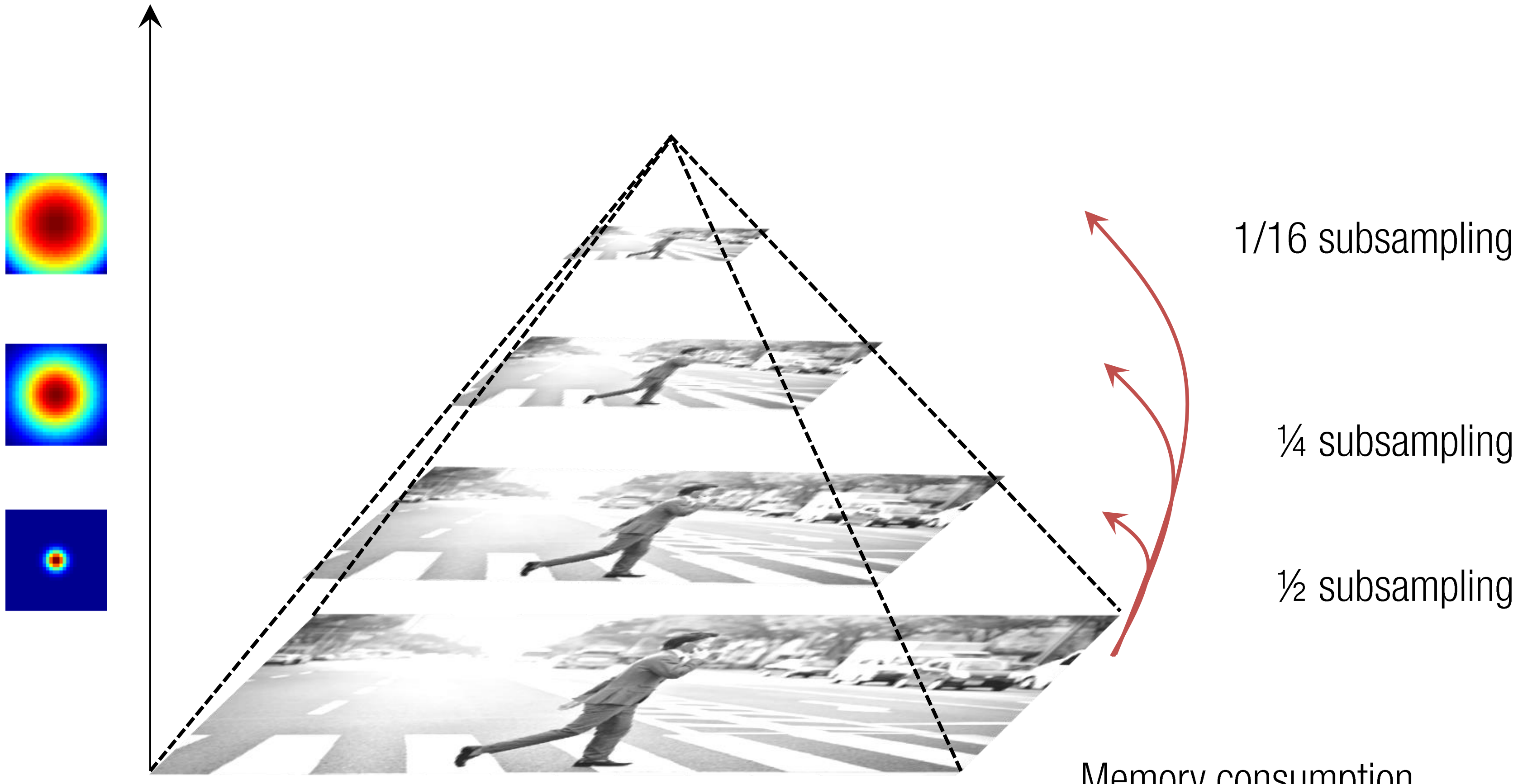
Gaussian Filtering and Then Subsampling

Image Reconstruction: Upsampling and Gaussian Blurring

Cf) Naïve Image Subsampling and Upsampling



Multi-dimensional Image Representation

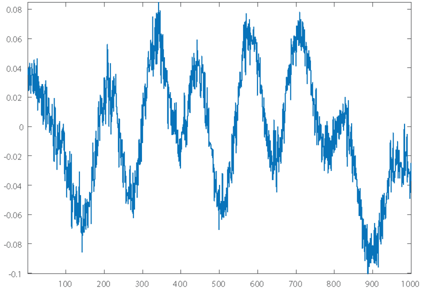


Memory consumption

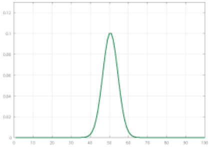
$$|I|(1+\frac{1}{4}+\frac{1}{16}+\cdots)=\frac{4}{3}|I|$$

GAUSSIAN IMAGE PYRAMID

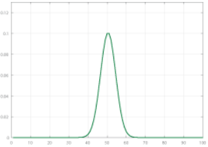
Composition of Gaussian Filters



*

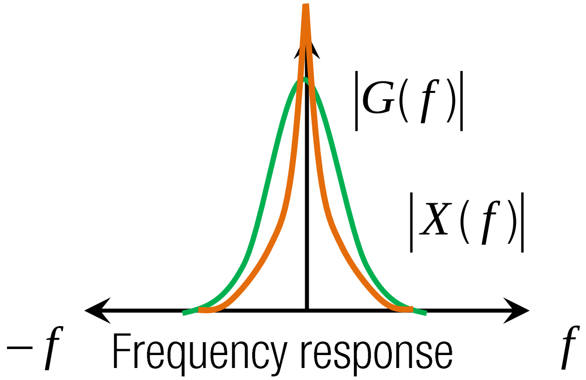


*



FT →

Inverse FT ←



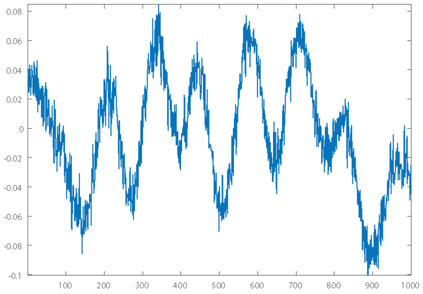
Time signal

Guassian filter

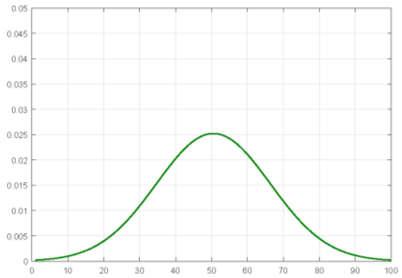
$x(t)$ $*$ $g(t;\sigma_1)$ $*$ $g(t;\sigma_2)$

$X(f)\,G(f)\,G(f)$

Composition of Gaussian Filters

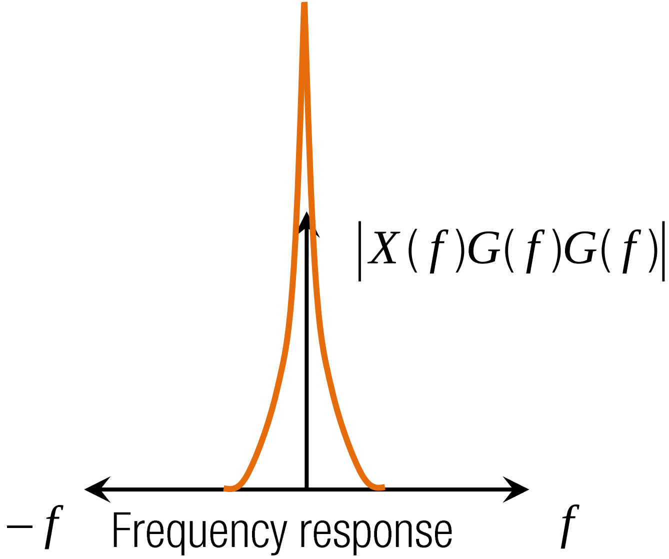


*



FT →

Inverse FT ←



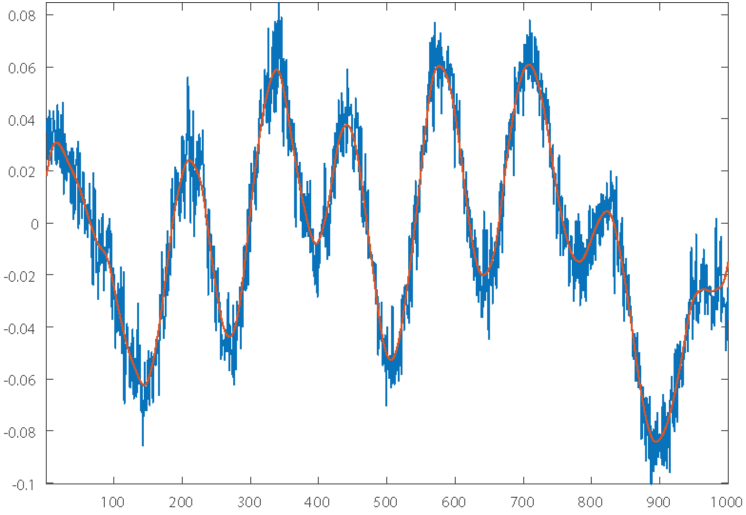
Time signal

Guassian filter

$x(t)$ $*$ $g(t;\sqrt{\sigma_1{}^2+\sigma_2{}^2})$

$X(f)\,G(f)\,G(f)$

Composition of Gaussian Filters

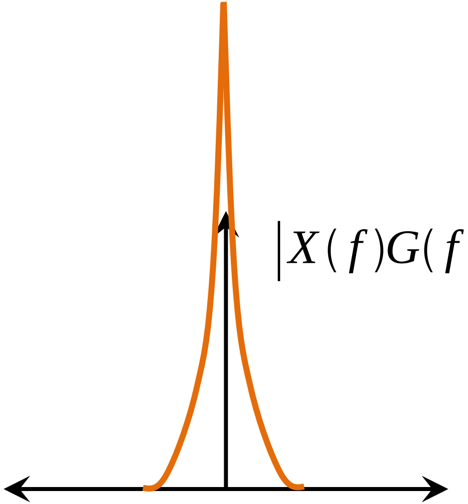


Time signal   Guassian filter

$$x(t) \quad * \quad g(t;\sqrt{\sigma_1{}^2+\sigma_2{}^2})$$

FT →

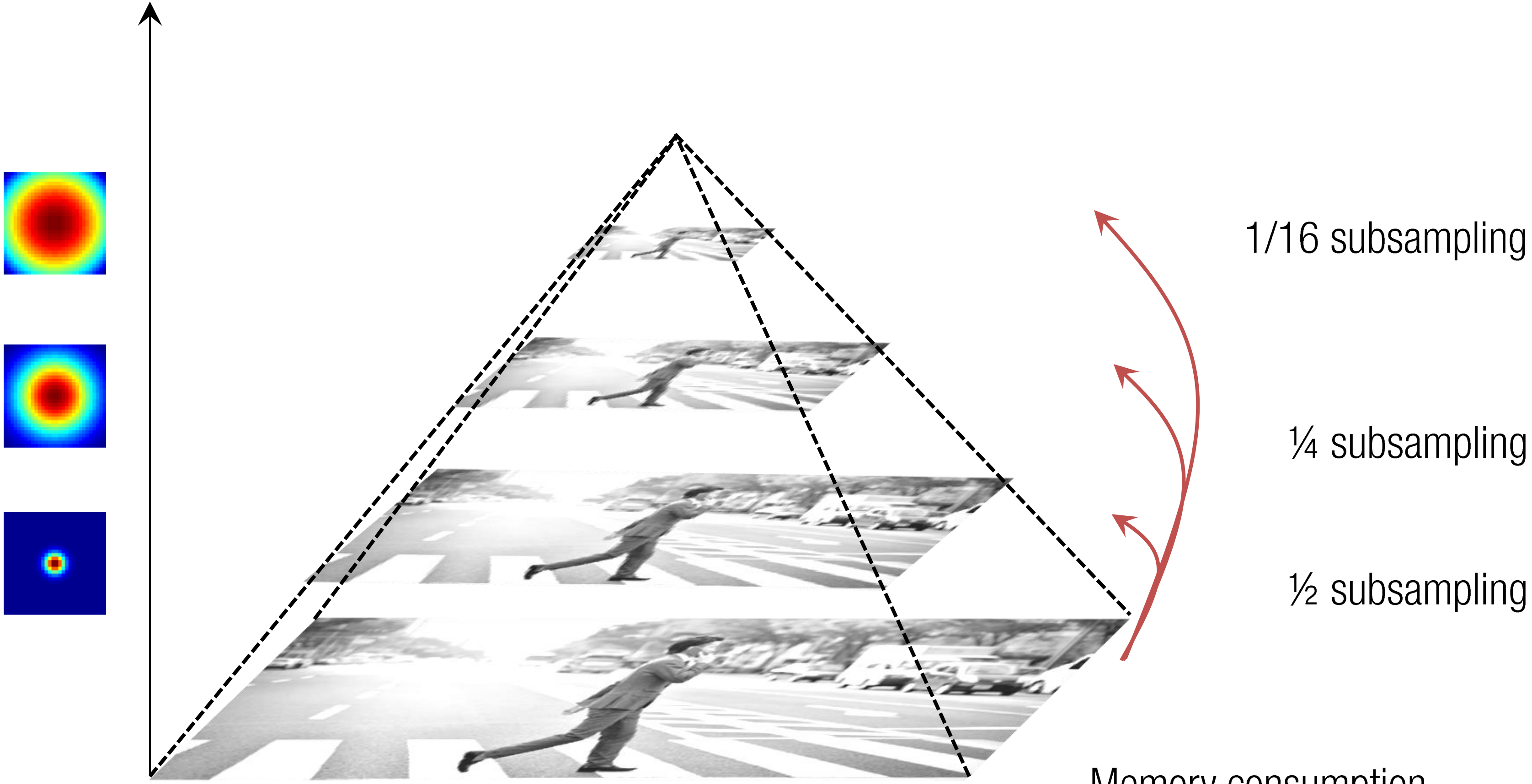
← Inverse FT



$|X(f)G(f)G(f)|$

$-f$ Frequency response f

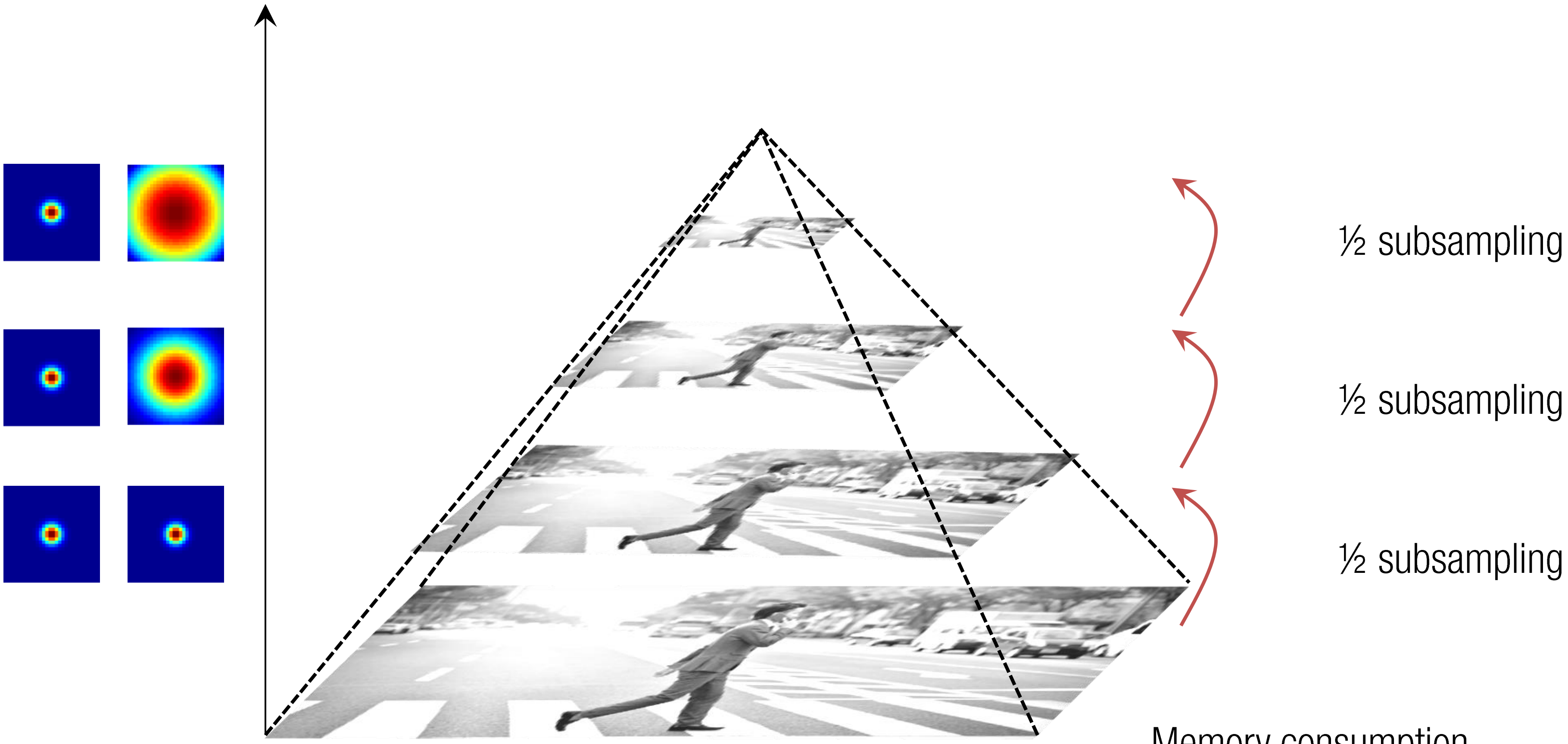
$$X(f)\,G(f)\,G(f)$$



GAUSSIAN IMAGE PYRAMID

Memory consumption

$$|I|\left(1+\frac{1}{4}+\frac{1}{16}+\cdots\right)=\frac{4}{3}|I|$$



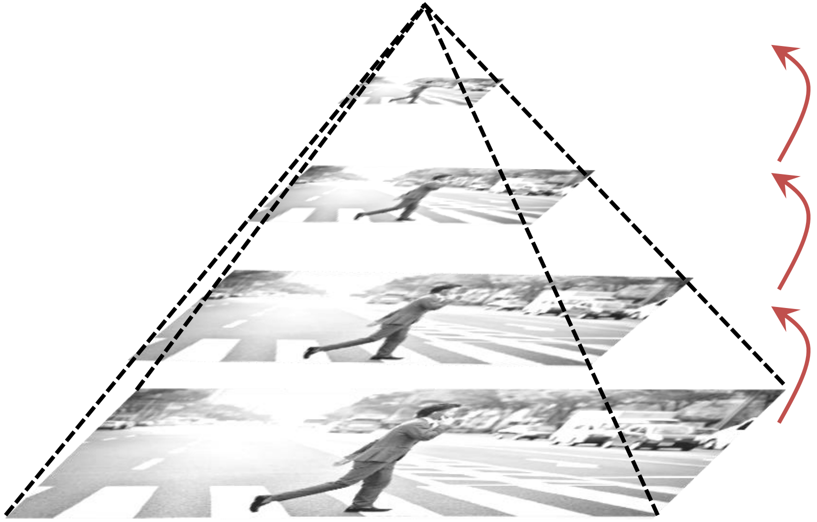
Memory consumption

$$|I|(1+\frac{1}{4}+\frac{1}{16}+\cdots)=\frac{4}{3}|I|$$

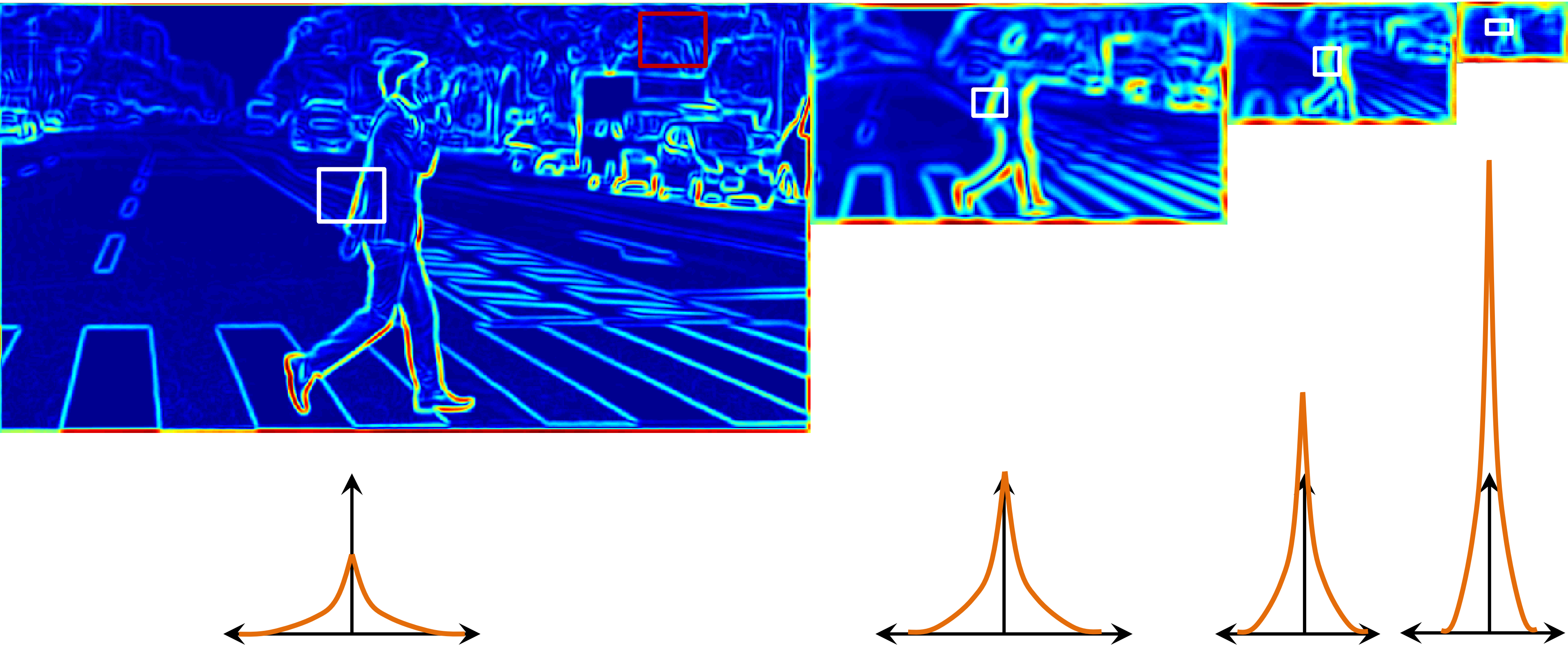
GAUSSIAN IMAGE PYRAMID

Gaussian Pyramid

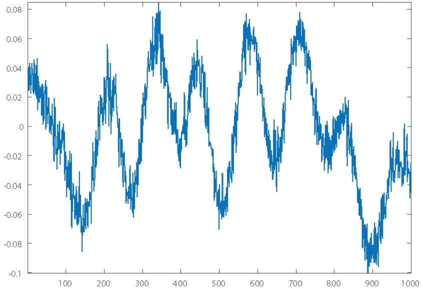
```
function im_pyramid = BuildGaussianPyramid(im, level)

im_pyramid{1} = im;

for i = 1 : level
    filter = fspecial('gaussian',10, sqrt(2));
    im_f = conv2(im_pyramid{i}, filter, 'same');
    im_pyramid{i+1} = imresize(im_f, 0.5, 'nearest');
end
```



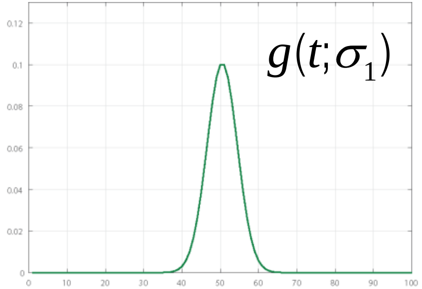
Redundant Representation of Gaussian Pyramid

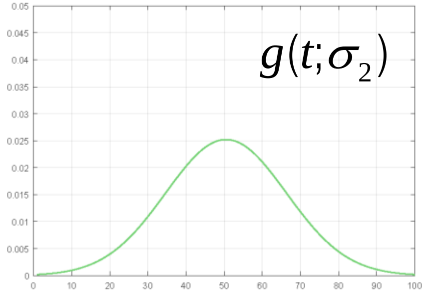


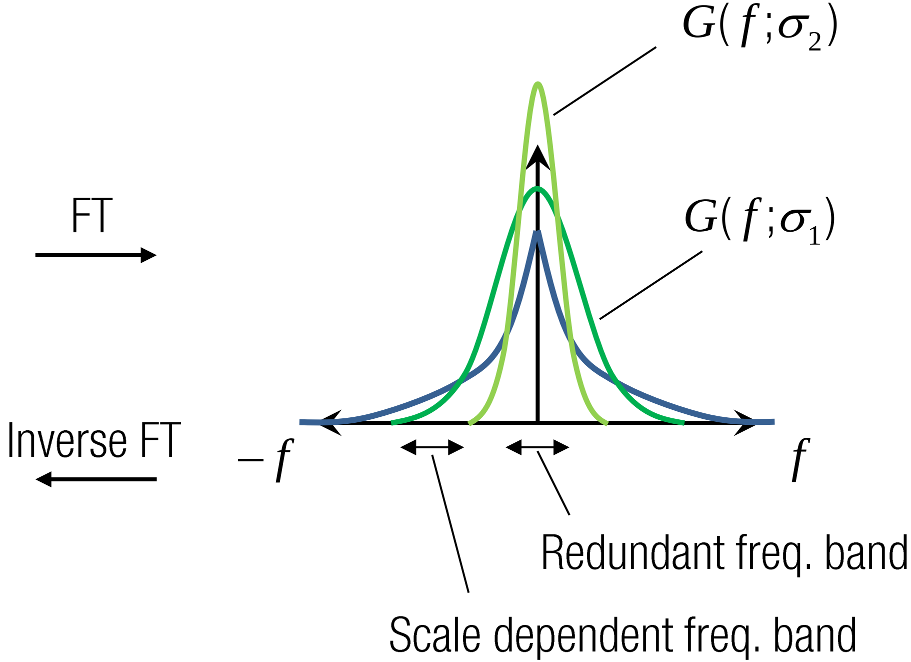
Fourier Transform



*

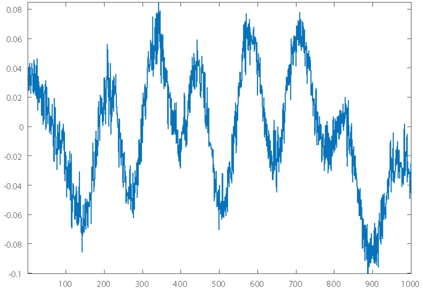


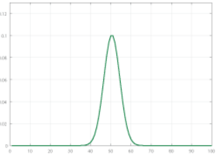
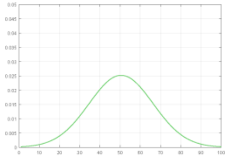




$x(t)$ * $g(t)$ $\sigma_1 < \sigma_2$ $X(f)\,G(f)$

Difference of Gaussian (DoG) ~ Band-pass Filter



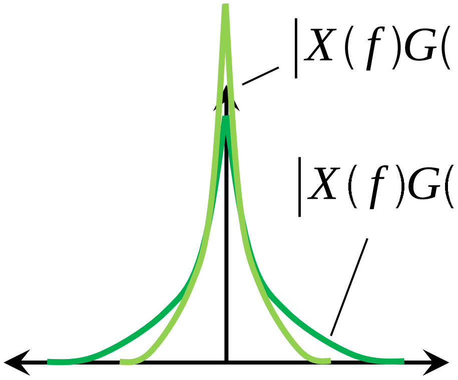
$*$ [ $g(t;\sigma_1)$ $-$  $g(t;\sigma_2)$]

FT →

← Inverse FT

$|X(f)G(f;\sigma_2)|$

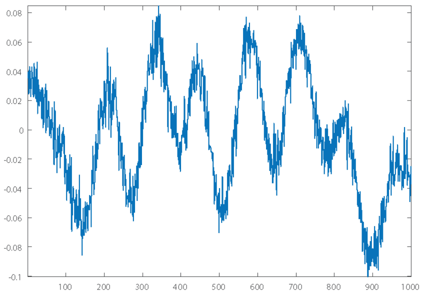
$|X(f)G(f;\sigma_1)|$



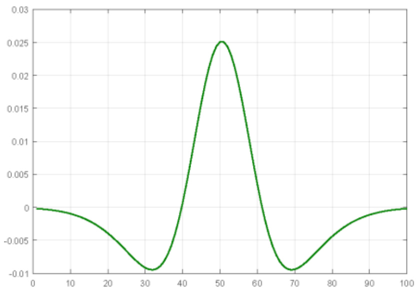
$-f$ f

$x(t) \quad * \quad g(t) \qquad\qquad X(f)\,G(f)$

Difference of Gaussian (DoG) ~ Band-pass Filter

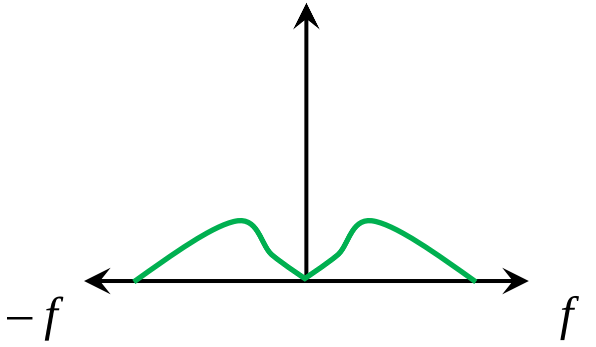


*



FT →

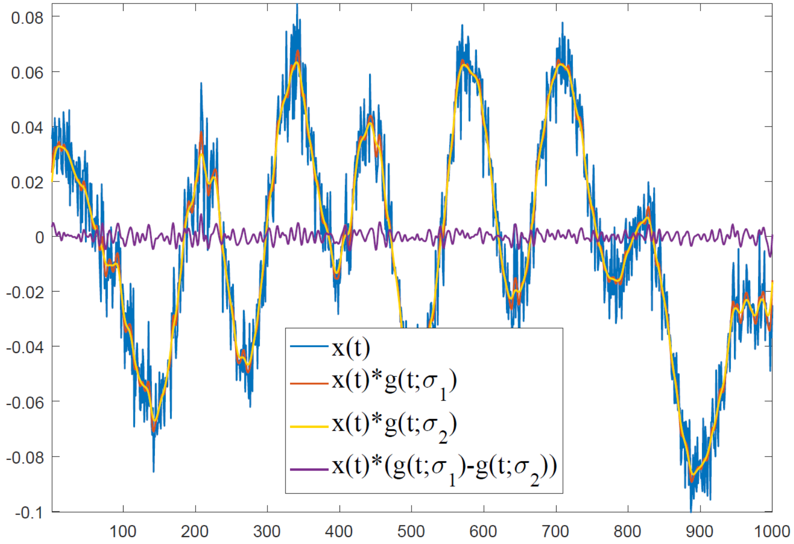
Inverse FT ←



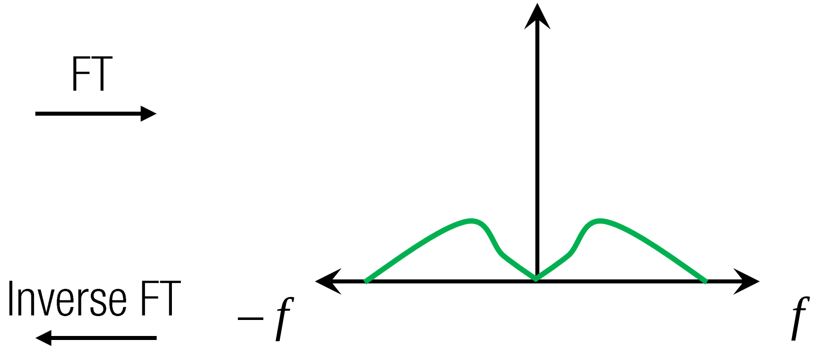
$x(t)$ $*$ $g(t;\sigma_1) - g(t;\sigma_2)$

$X(f)\big(G(f;\sigma_1) - G(f;\sigma_2)\big)$

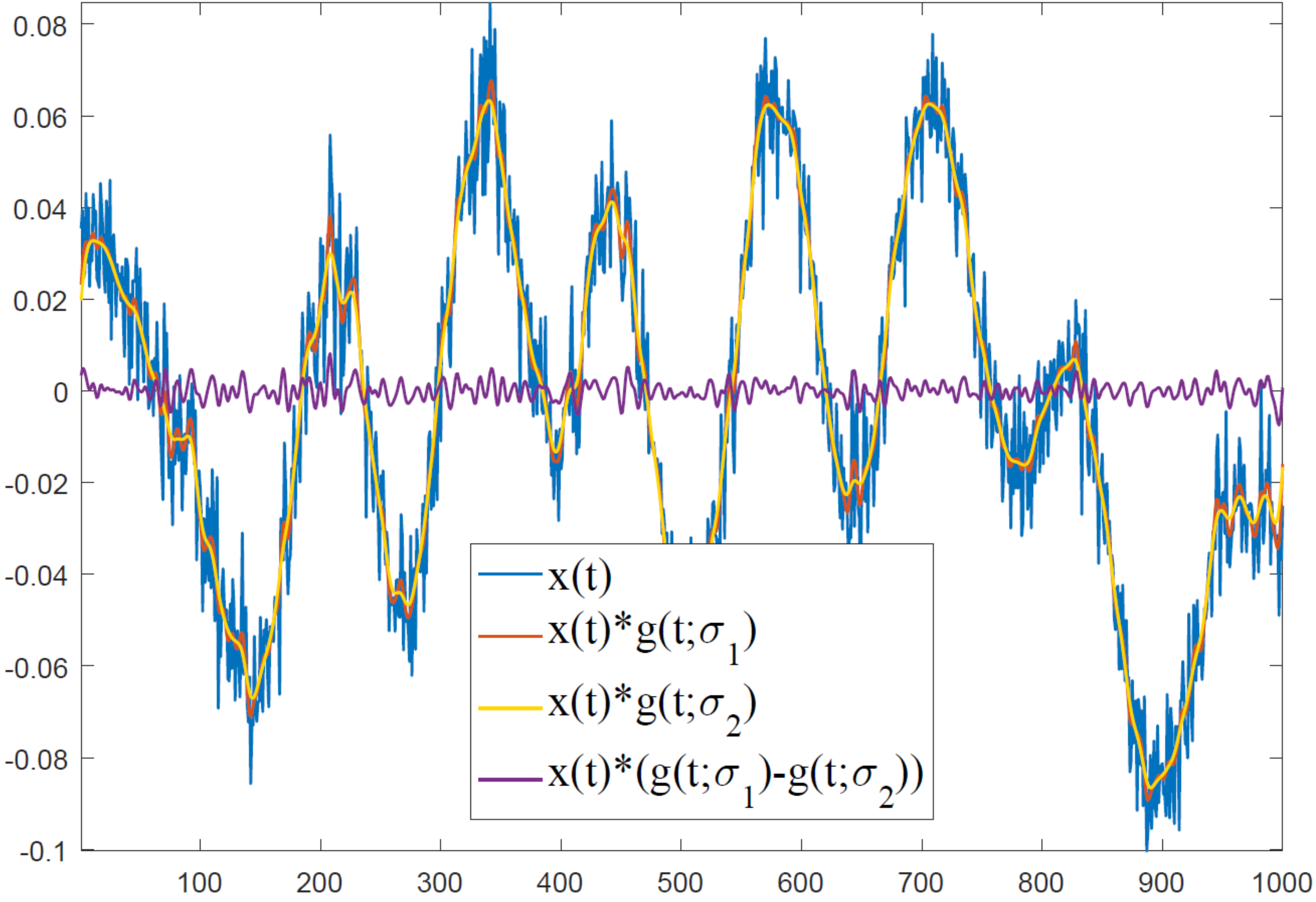
Difference of Gaussian (DoG) ~ Band-pass Filter



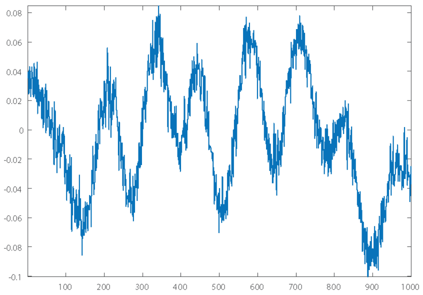
$$x(t) \quad * \quad g(t;\sigma_1) - g(t;\sigma_2)$$



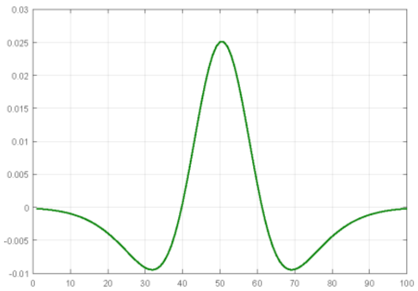
$$X(f)\big(G(f;\sigma_1) - G(f;\sigma_2)\big)$$


0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.08
-0.1
100
200
300
400
500
600
700
800
900
1000
x(t)
x(t)*g(t;σ1)
x(t)*g(t;σ2)
x(t)*(g(t;σ1)-g(t;σ2))

Difference of Gaussian (DoG) ~ Band-pass Filter

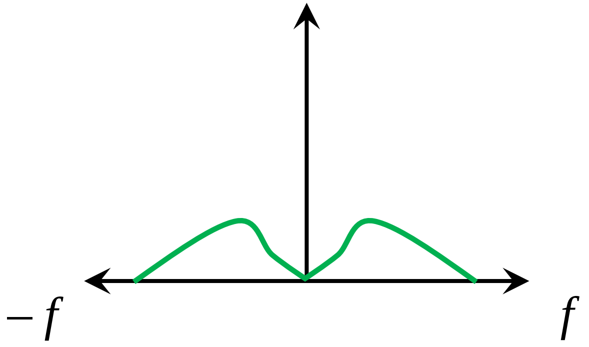


*



FT →

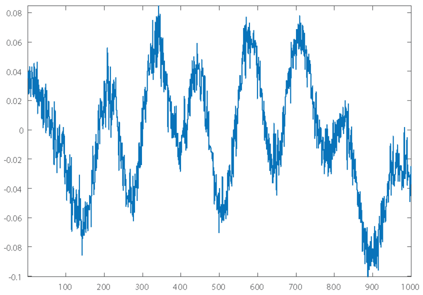
← Inverse FT



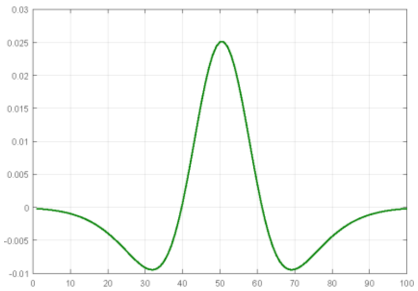
$x(t)$ $*$ $g(t;\sigma_1) - g(t;\sigma_2)$

$X(f)\left(G(f;\sigma_1) - G(f;\sigma_2)\right)$

Laplacian of Gaussian (LoG) ~ DoG

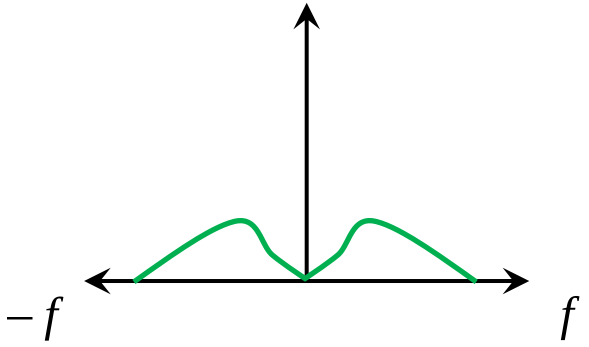


$*$



FT →

Inverse FT ←



$x(t)$ $*$ $g(t;\sigma_1) - g(t;\sigma_2)$

$\approx \nabla \cdot \nabla g$

Laplacian of Gaussian

$X(f)\big(G(f;\sigma_1) - G(f;\sigma_2)\big)$

Laplacian of Gaussian (LoG) ~ DoG

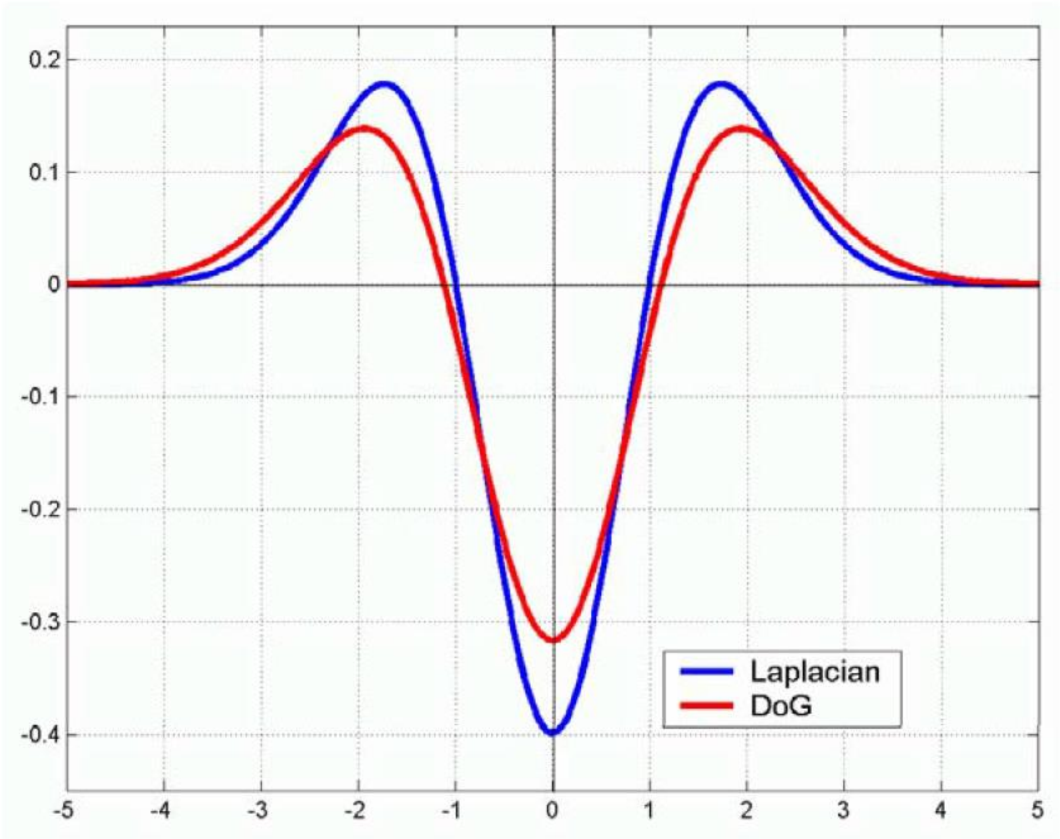
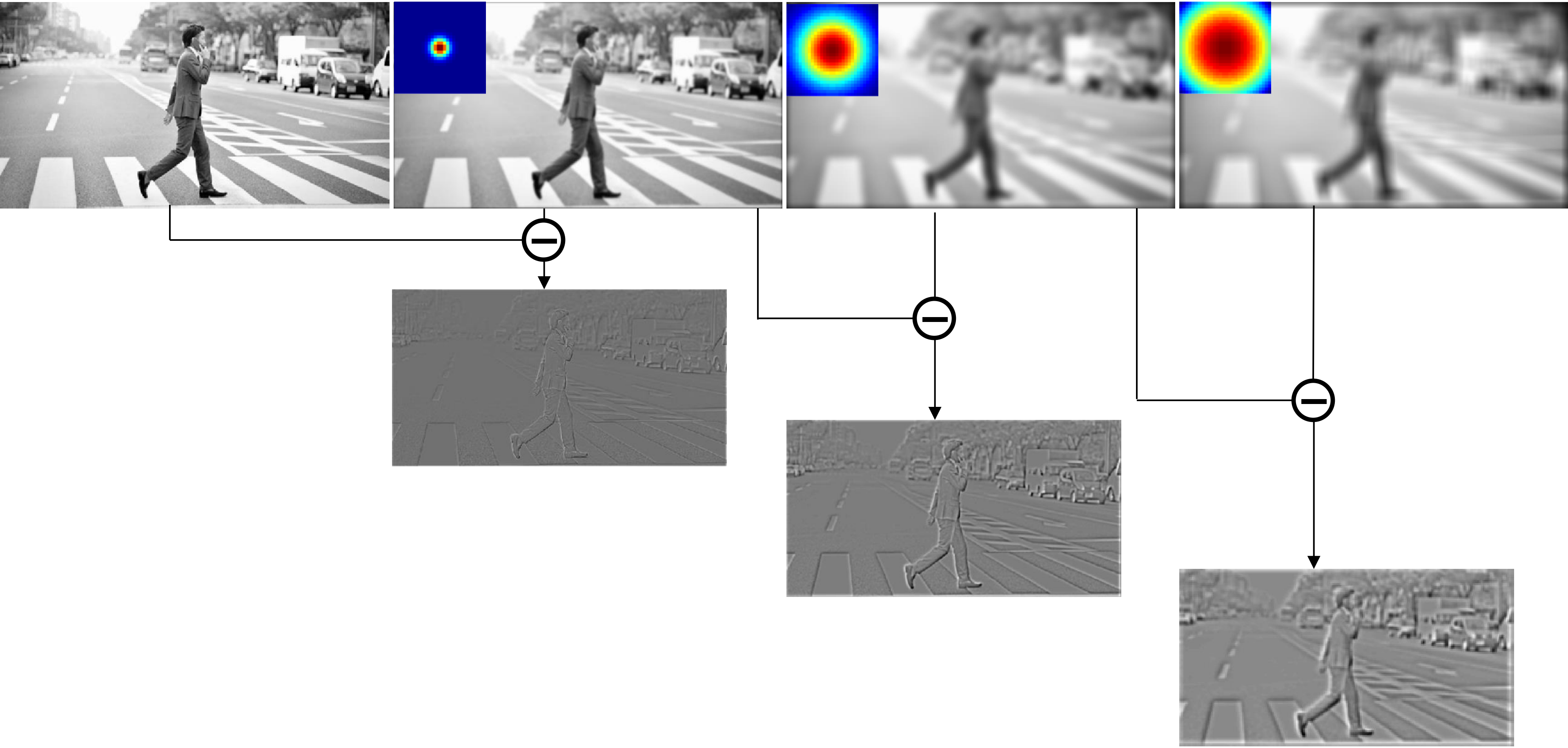
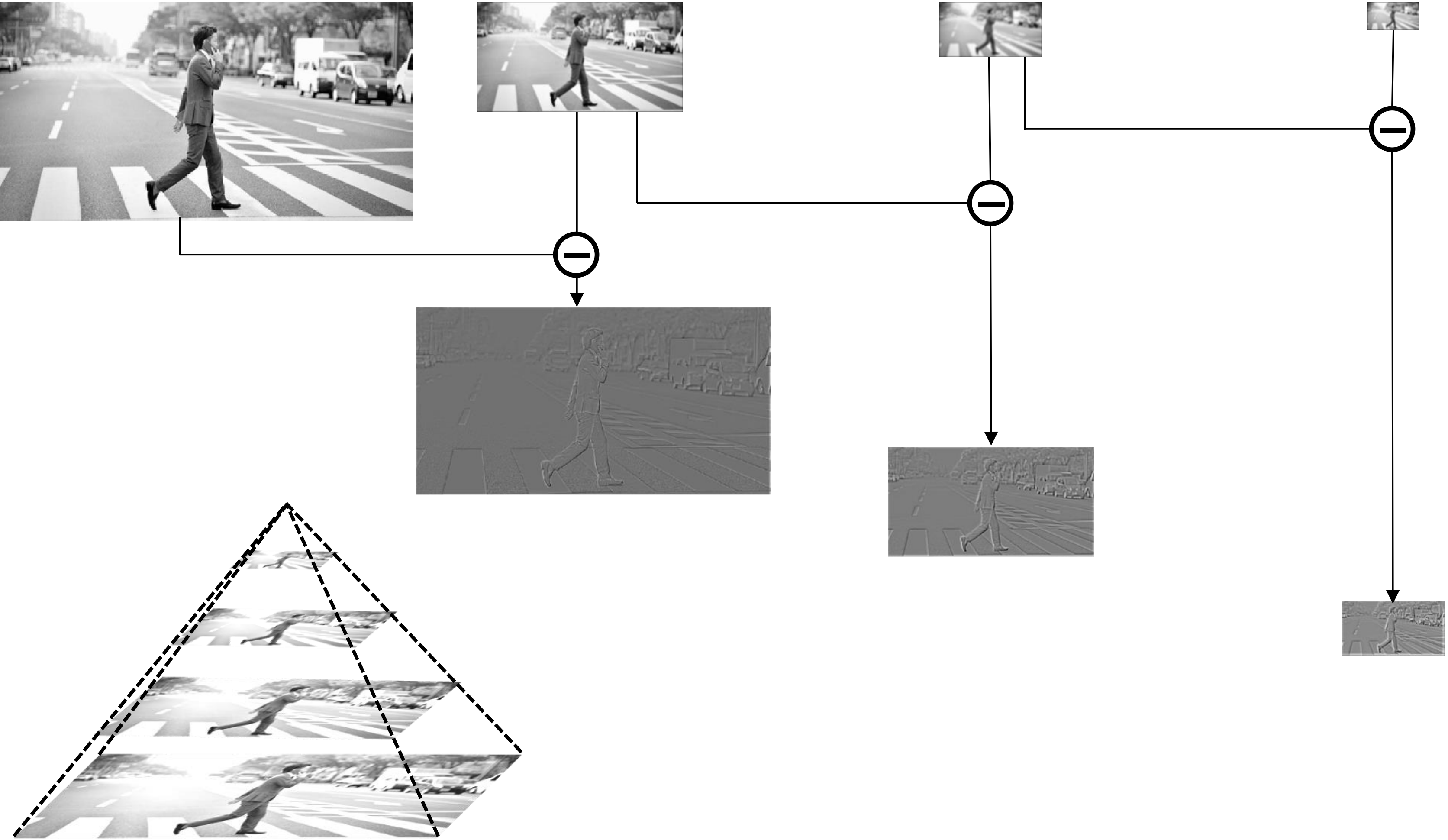
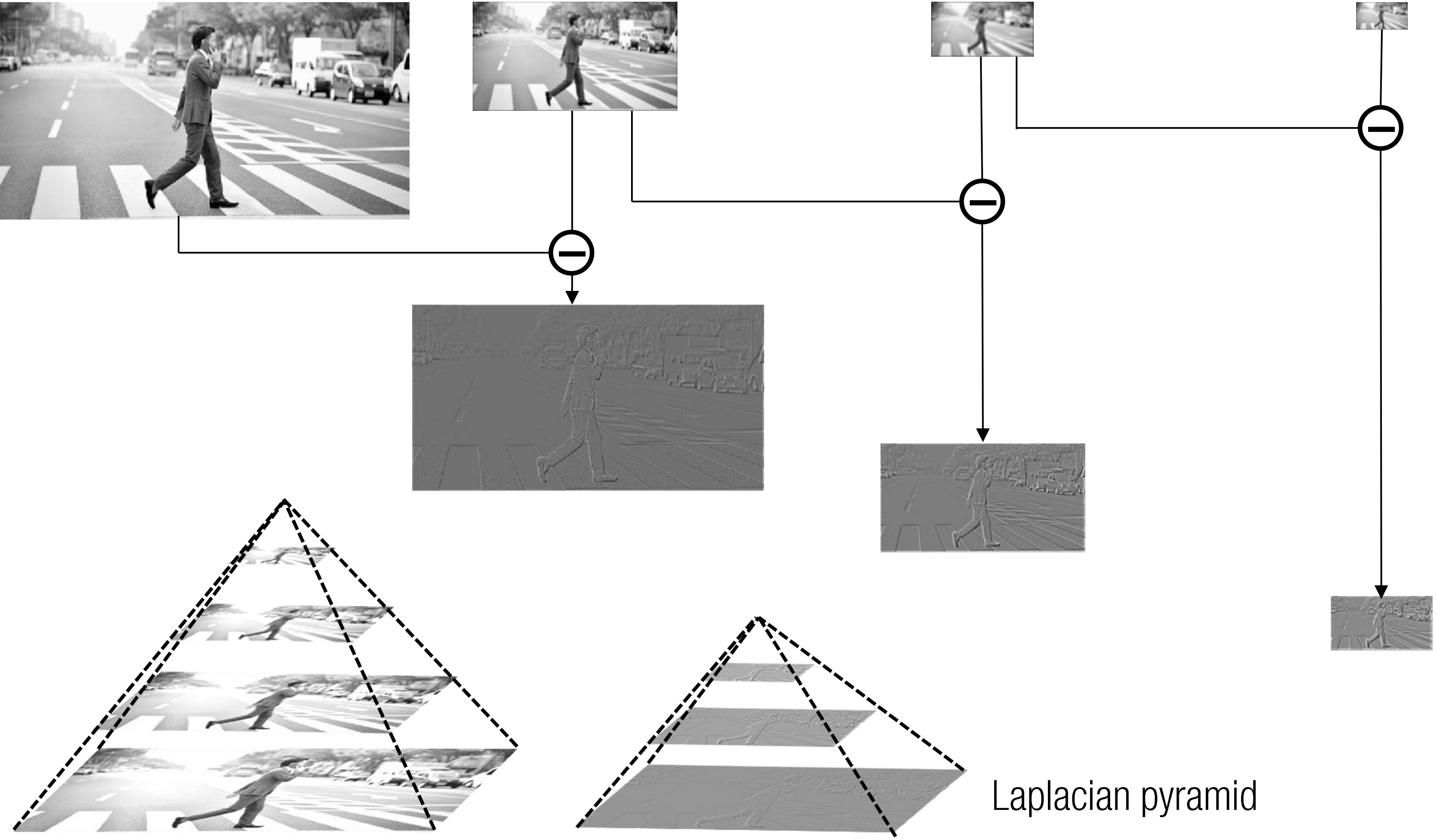


IMAGE LAPLACIAN

IMAGE LAPLACIAN

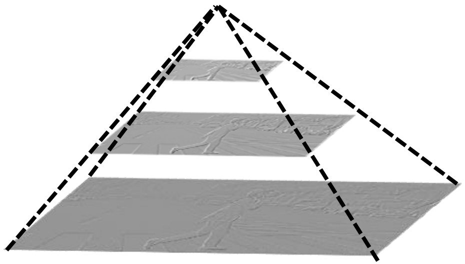


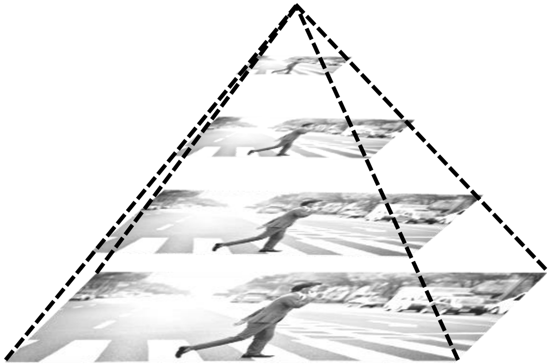



Laplacian pyramid

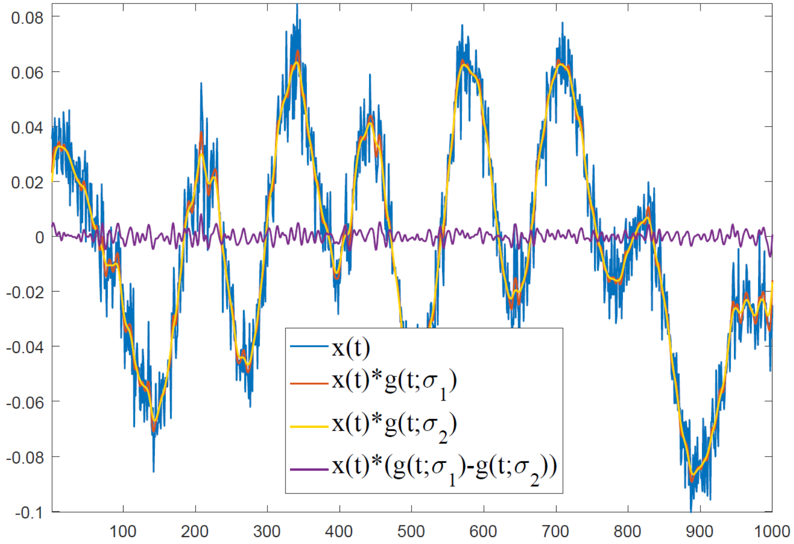
LAPLACIAN PYRAMID

```
function [g_pyramid, l_pyramid] = BuildLaplacianPyramid(im, level)

g_pyramid{1} = im;

for i = 1 : level
    filter = fspecial('gaussian',10, sqrt(2));
    im_f = conv2(g_pyramid{i}, filter, 'same');
    l_pyramid{i} = double(g_pyramid{i})-im_f;
    g_pyramid{i+1} = imresize(im_f, 0.5, 'nearest');
end
```

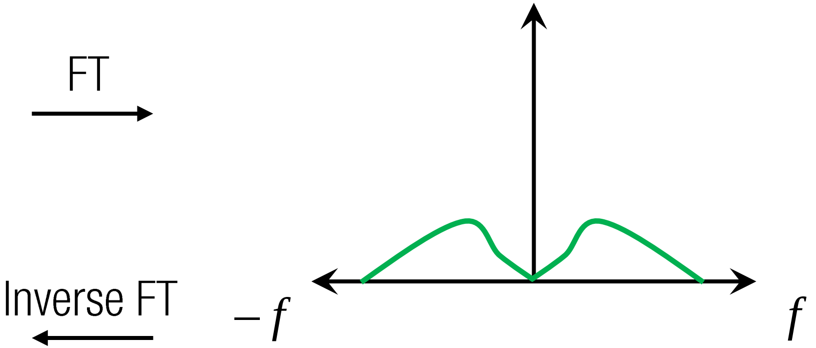




Difference of Gaussian (DoG) ~ Band-pass Filter

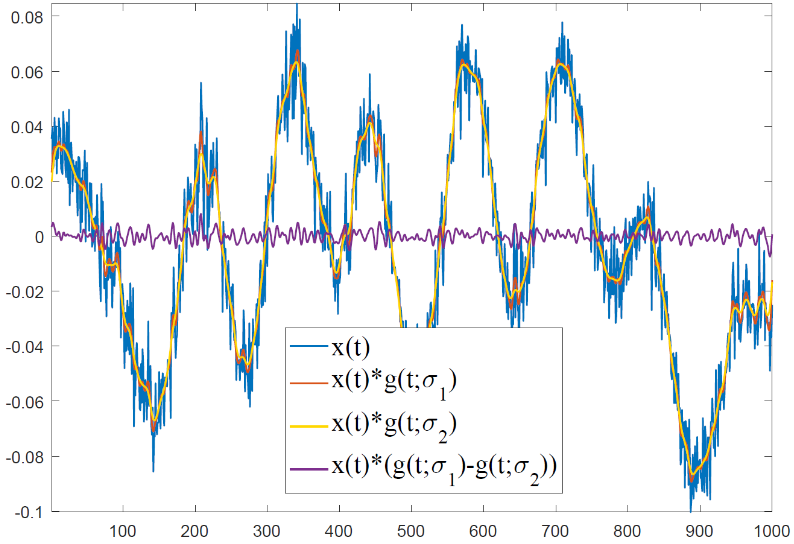


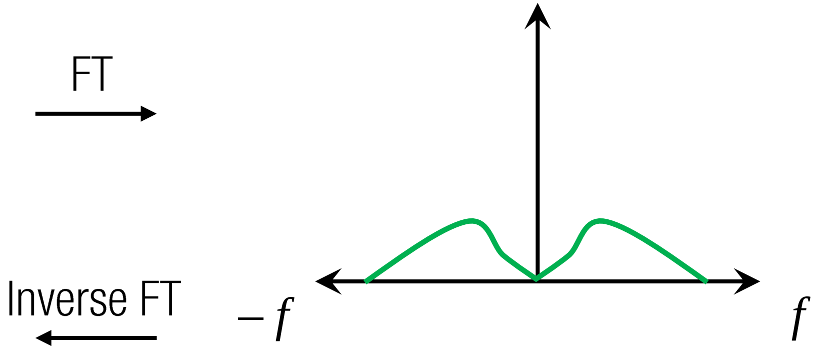
$x(t) \quad * \quad g(t;\sigma_1) - g(t;\sigma_2)$



$X(f)\big(G(f;\sigma_1) - G(f;\sigma_2)\big)$

Signal Reconstruction ~ LoG + G. Filtering





$$l(t) = x(t) * \left(g(t;\sigma_1) - g(t;\sigma_2)\right)$$

$$x(t) * g(t;\sigma_1) = l(t) + x(t) * g(t;\sigma_2)$$

Signal reconstruction with laplacian

$$L(f) = X(f)\left(G(f;\sigma_1) - G(f;\sigma_2)\right)$$

$$X(f)G(f;\sigma_1) = \underbrace{L(f)}_{\text{LoG}} + \underbrace{X(f)G(f;\sigma_2)}_{\text{Smoother signal}}$$

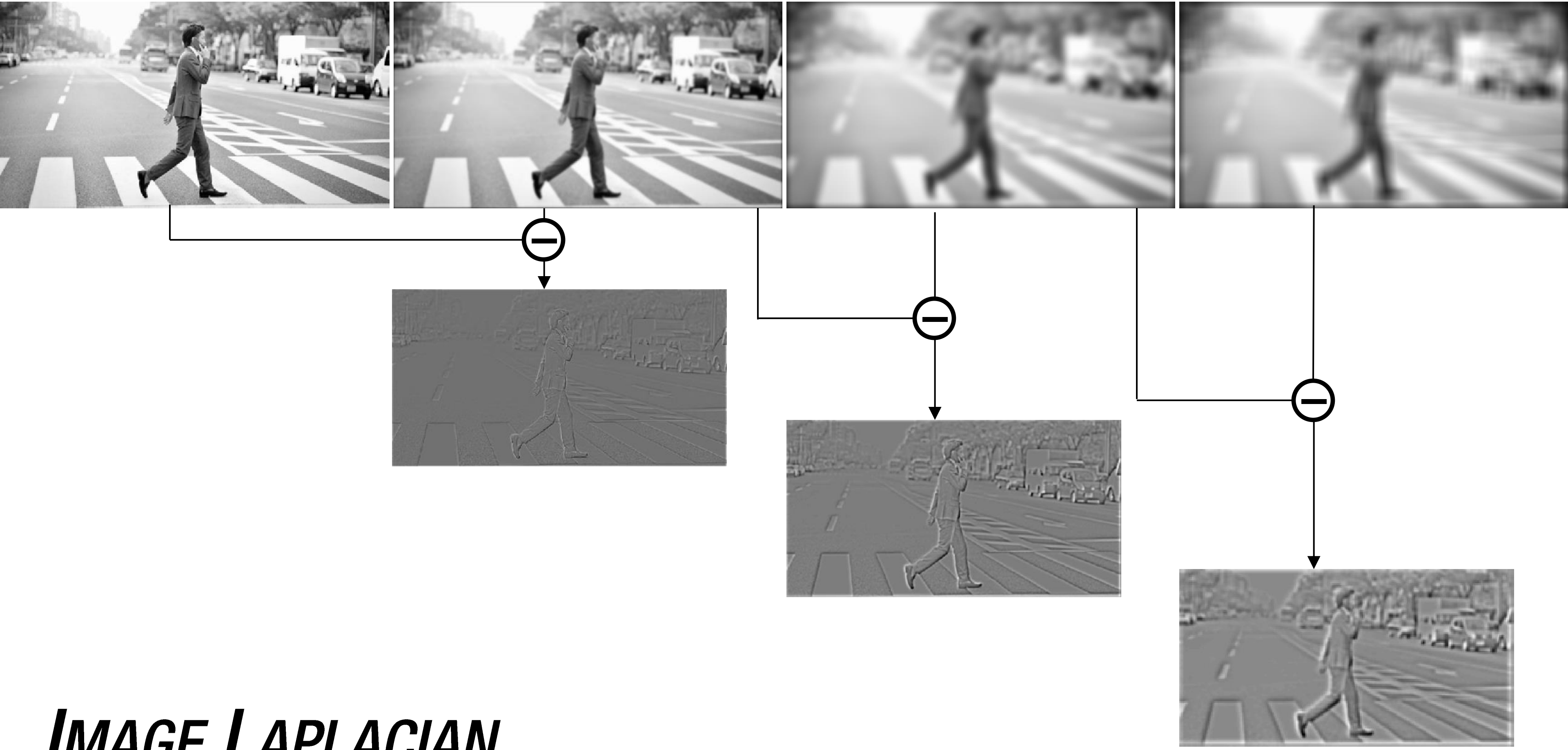


IMAGE LAPLACIAN

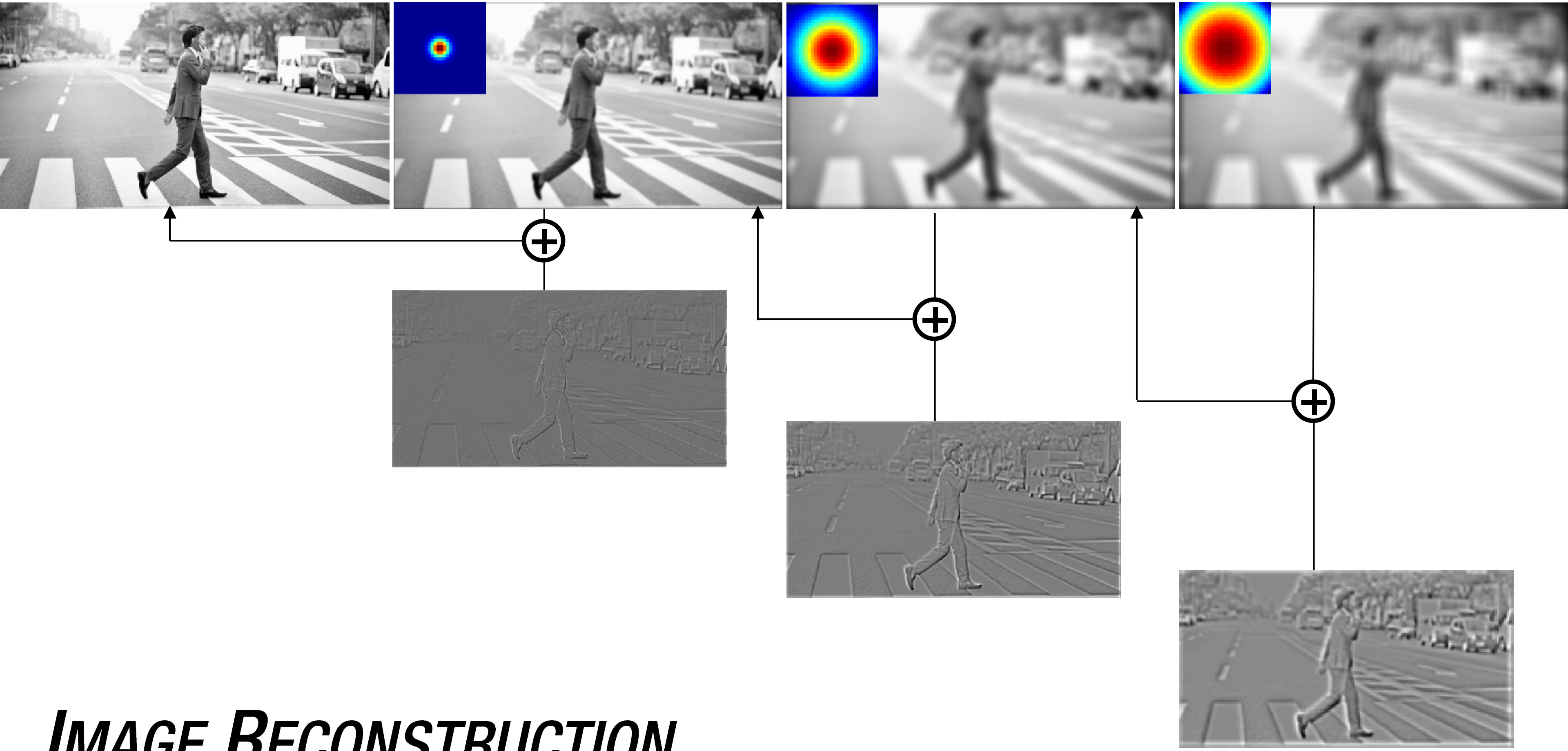
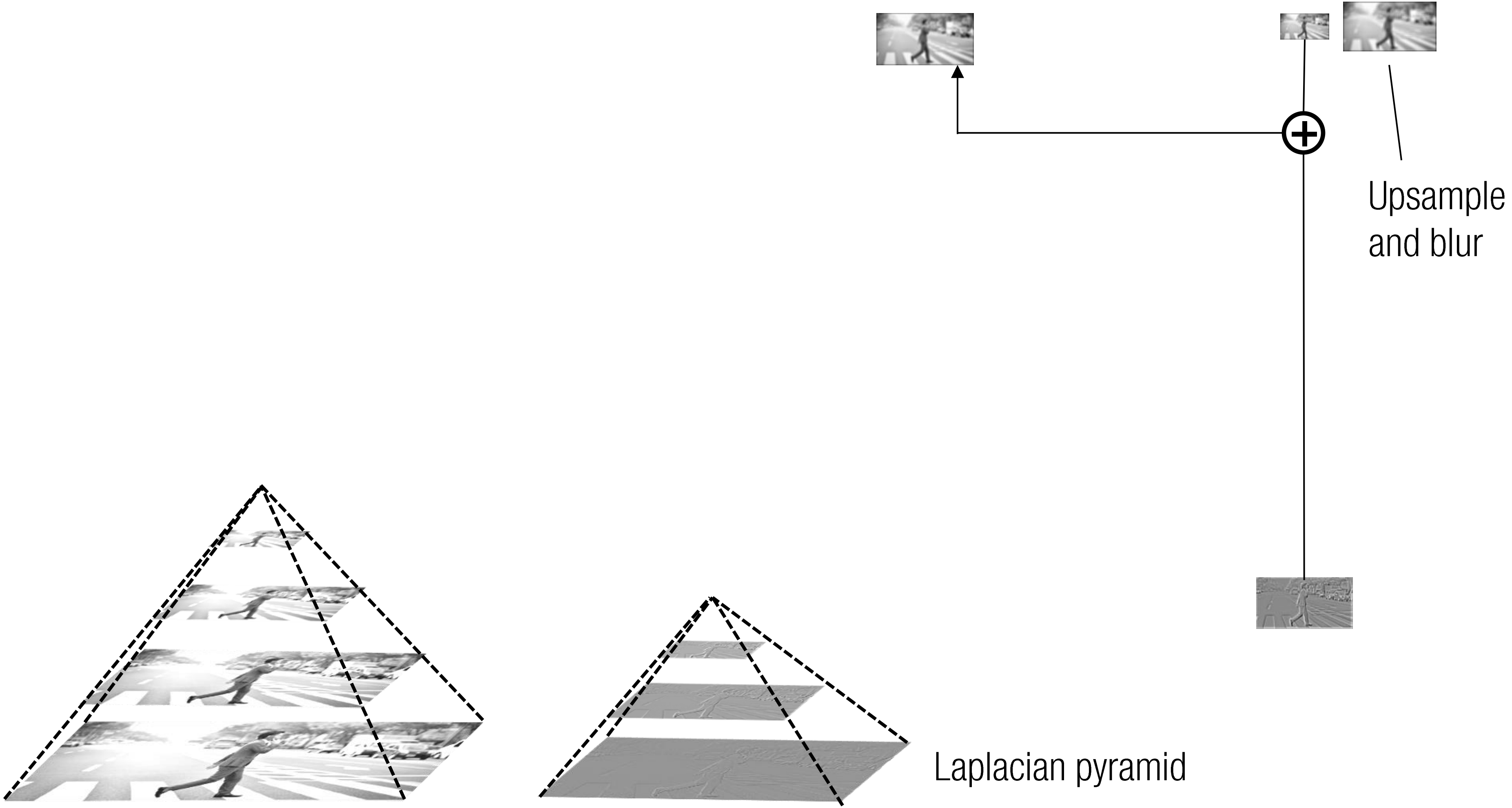
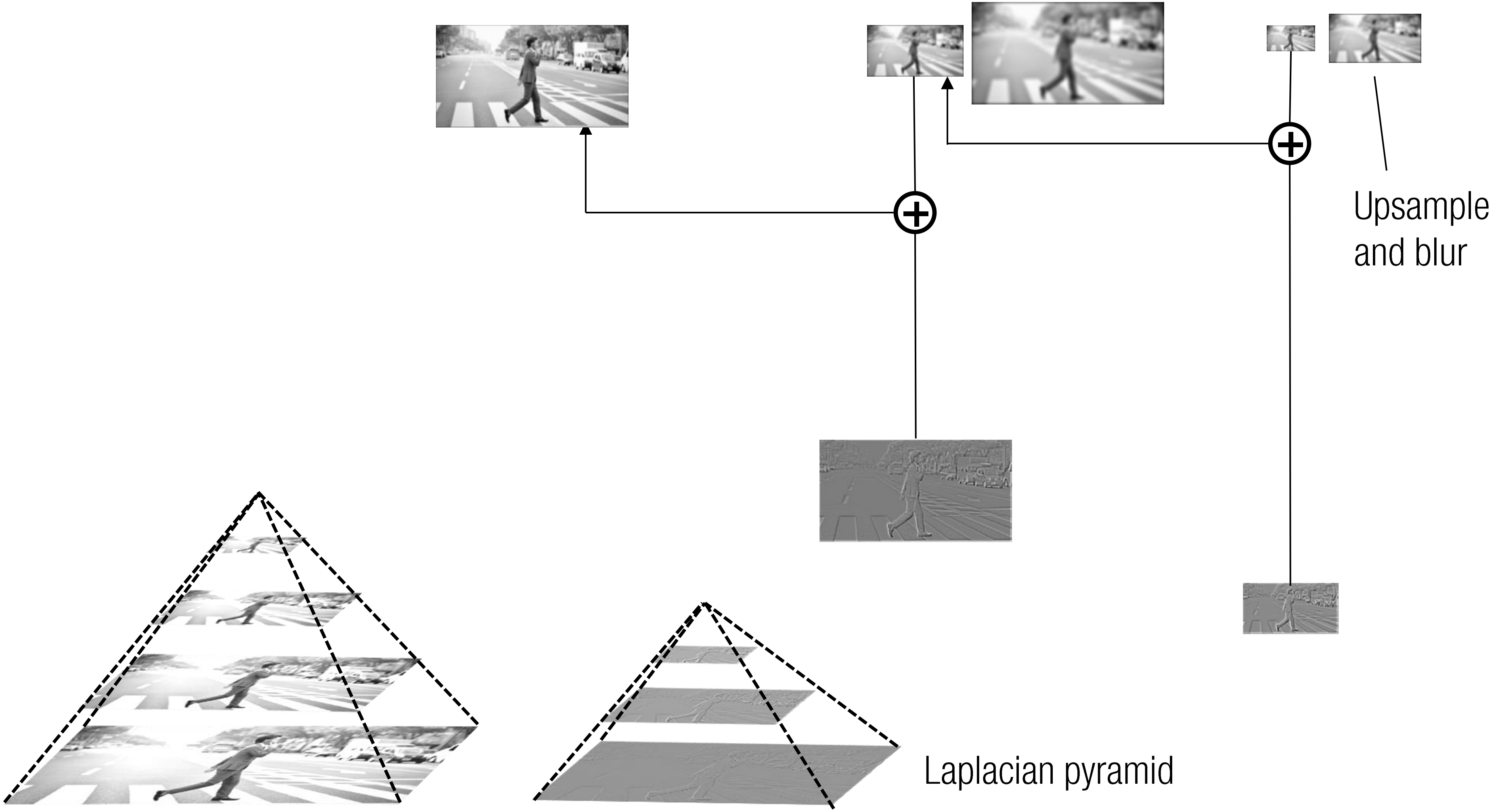
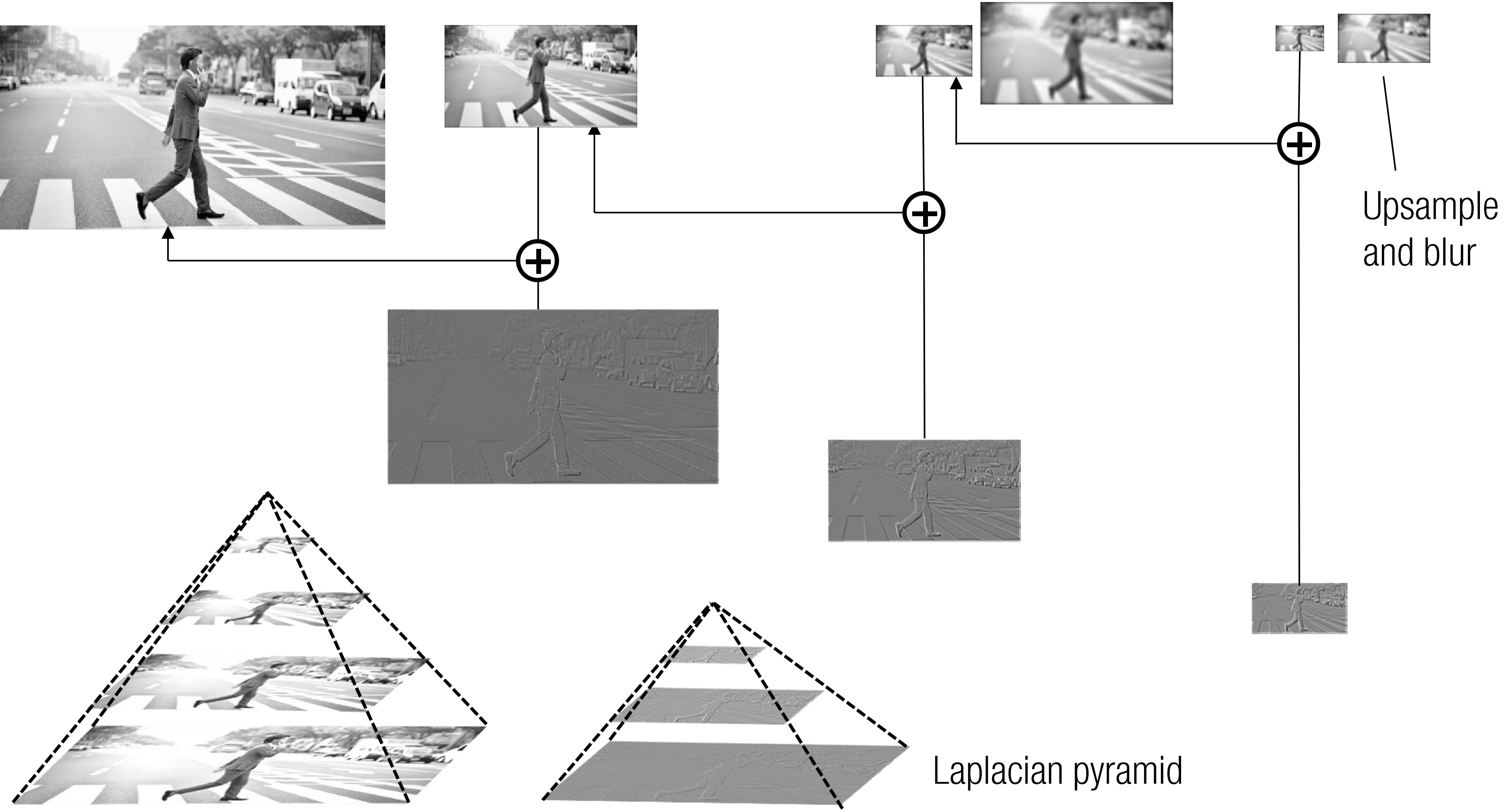


IMAGE RECONSTRUCTION


Upsample
and blur
Laplacian pyramid



Upsample
and blur
Laplacian pyramid



Upsample
and blur
Laplacian pyramid