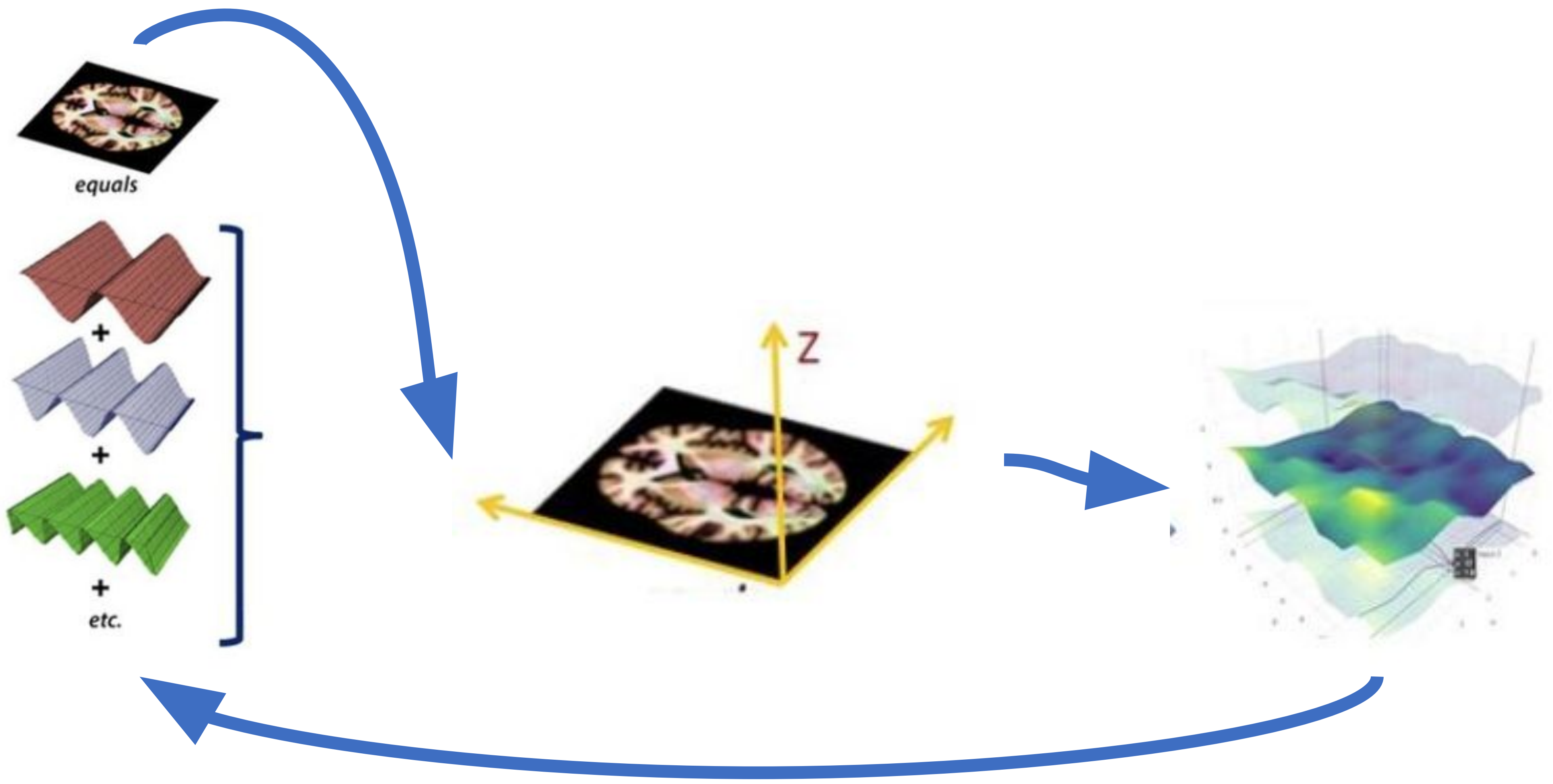
Discussion 3 – Pyramids & FFT

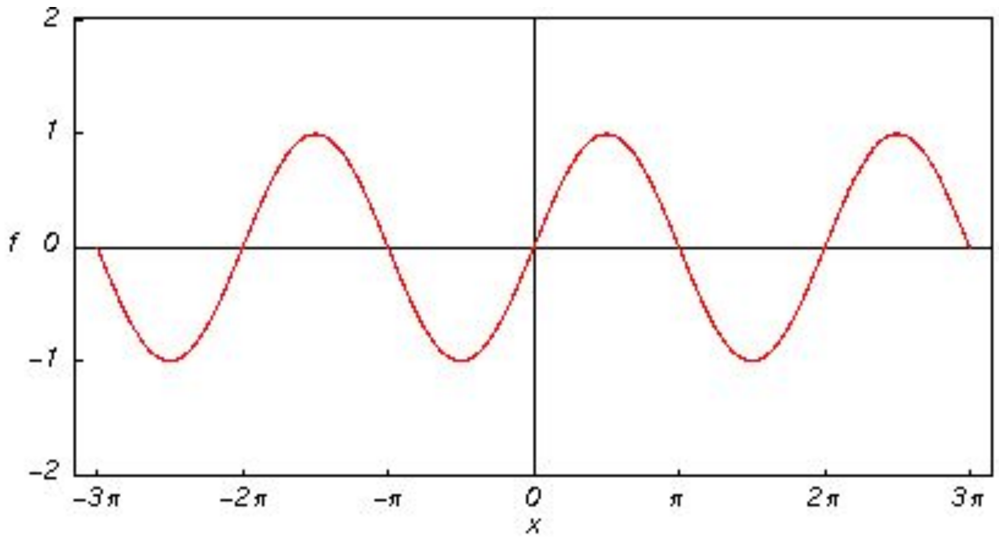
EECS 442 - Fall 2023

2D Fourier Transform

2D Fourier Transform



2D Fourier Transform



- Amplitude
- Frequency
- Phase
- Direction

Pairs of Fourier Transforms

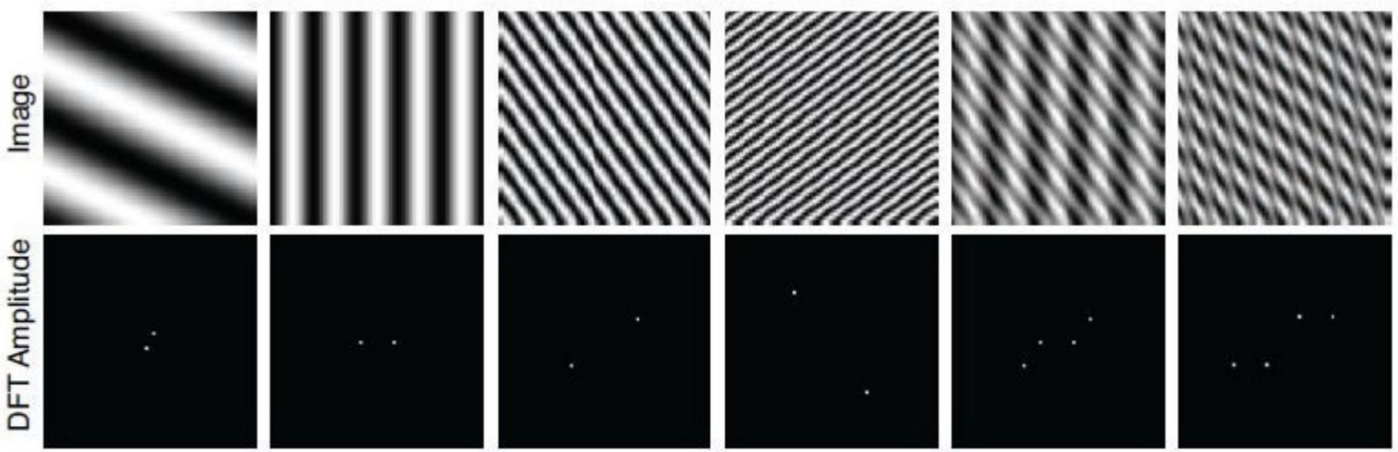


Figure 2.6: Some two-dimensional Fourier transform pairs. Images are 64 × 64 pixels. The waves are *cos* with frequencies (1, 2), (5, 0), (10, 7), (11, −15). The last two examples show the sum of two waves and the product.

$$X(m) = \sum_{n=0}^{N-1} x(n)[\cos(2\pi nm/N) - i\sin(2\pi nm/N)]$$

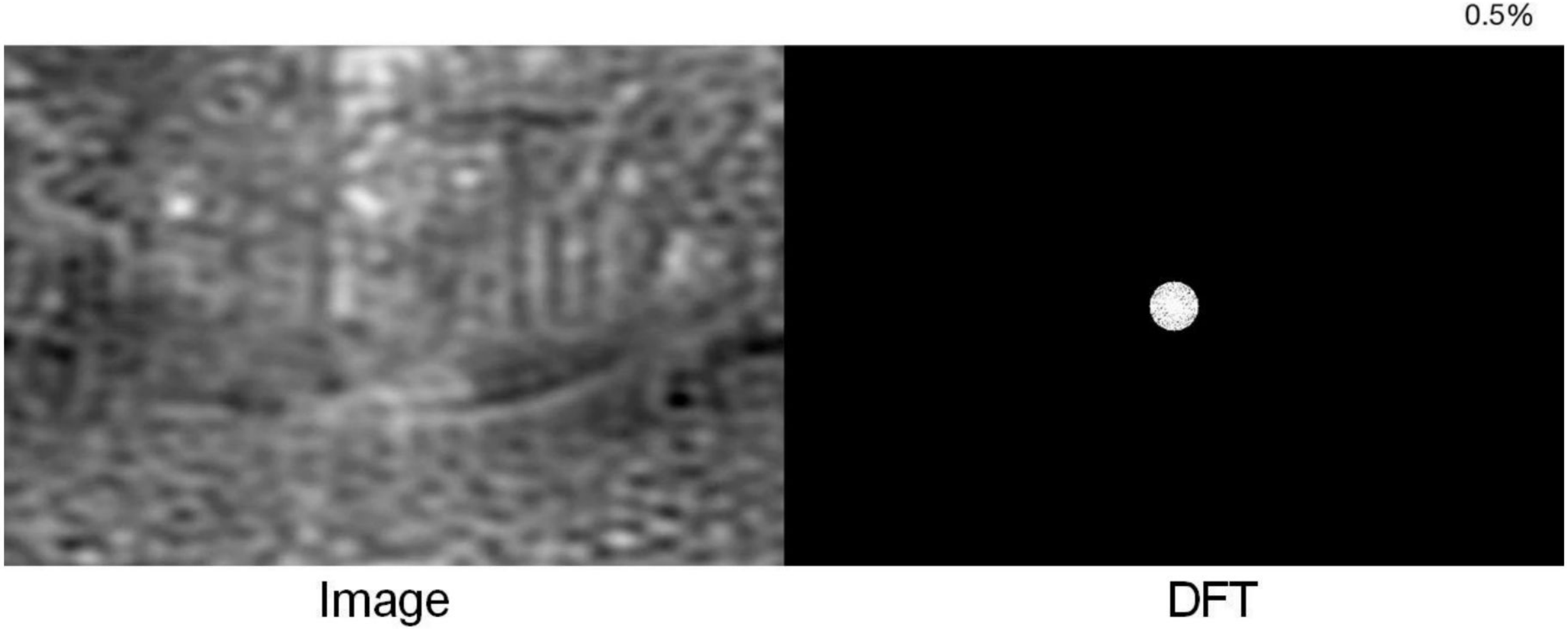
- X(m): m^th output of DFT, e.g. , X(0), X(1), ..., X(m)
- x(n): Input samples, e.g. , x(0), x(1), ..., x(n)
- i: Imaginary numbers line
- N: length of Input

$$X_{\text{mag}}(m) = |X(m)| = \sqrt{X_{\text{ real }}(m)^2 + X_{\text{ inag }}(m)^2}$$

$$X_\phi(m) = \tan^{-1}\left(\frac{X_{\text{imag}}(m)}{X_{\text{real}}(m)}\right)$$

f(m)= $\frac{mf_s}{N}$

Reconstruct an image, low frequency to high



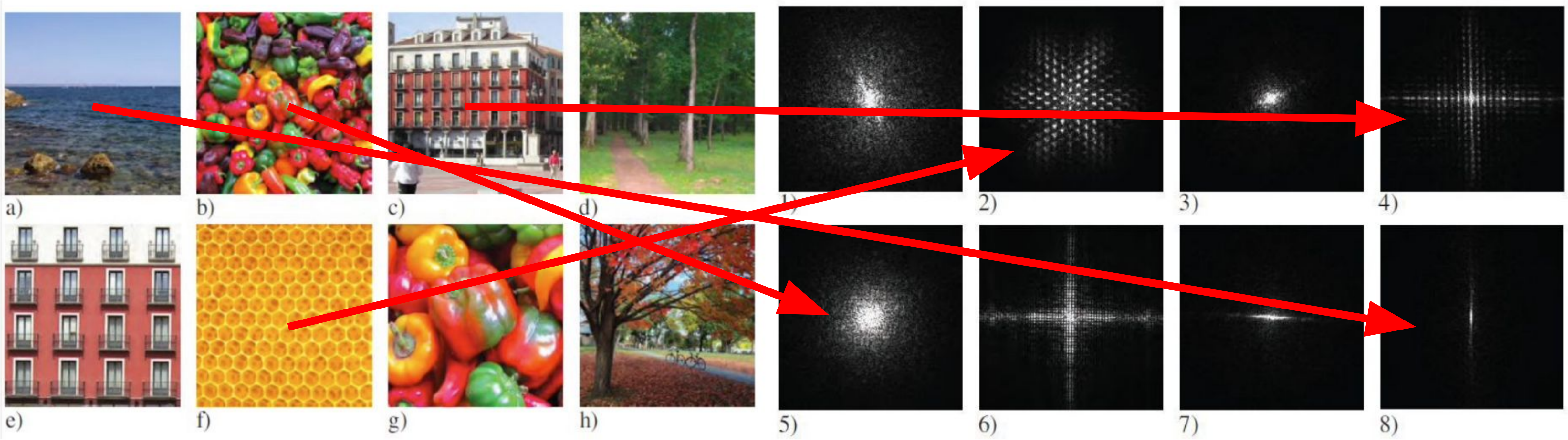
Reconstruct an image, low frequency to high

4.6%

Image DFT

Reconstruct an image, low frequency to high

Fourier Matching Game



Match each image (a-h) with its corresponding Fourier transform magnitude (1-8)

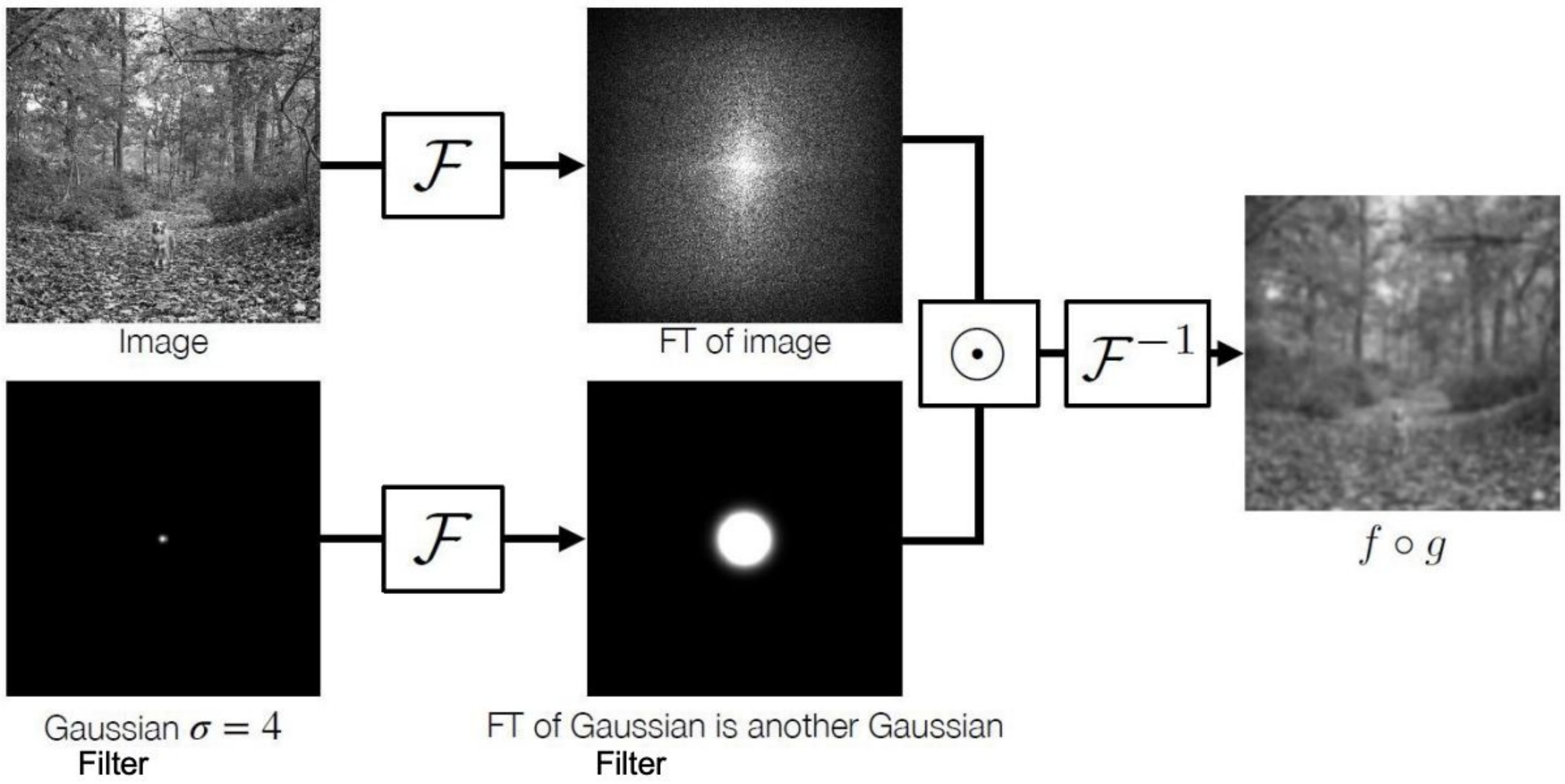
Convolution Theorem of Fourier Transform

- 2D Fourier transform is separable (just like Gaussian)
- Computable in *O(nlogn)* (using FFT)
- Convolution Theorem: convolution is pointwise multiplication in the Fourier domain!

$$\mathcal{F}\{f \circ g\} = \mathcal{F}\{f\} \odot \mathcal{F}\{g\}$$

- Useful trick for fast convolutions, especially for large filters

Convolution Theorem Example



Convolution Theorem – Why it matters?

Conv in space domain

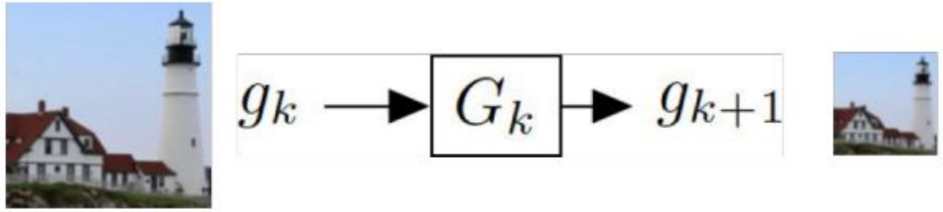
- Convolve the whole image with a filter
- Expensive to compute
- O(n^4) for 2D convolution

Conv in frequency domain

- FFT + Pointwise multiplications
- Much faster to compute
- O(n^2 log^2(n)) for 2D FFT

Gaussian & Laplacian Pyramids

Gaussian Pyramid Logic

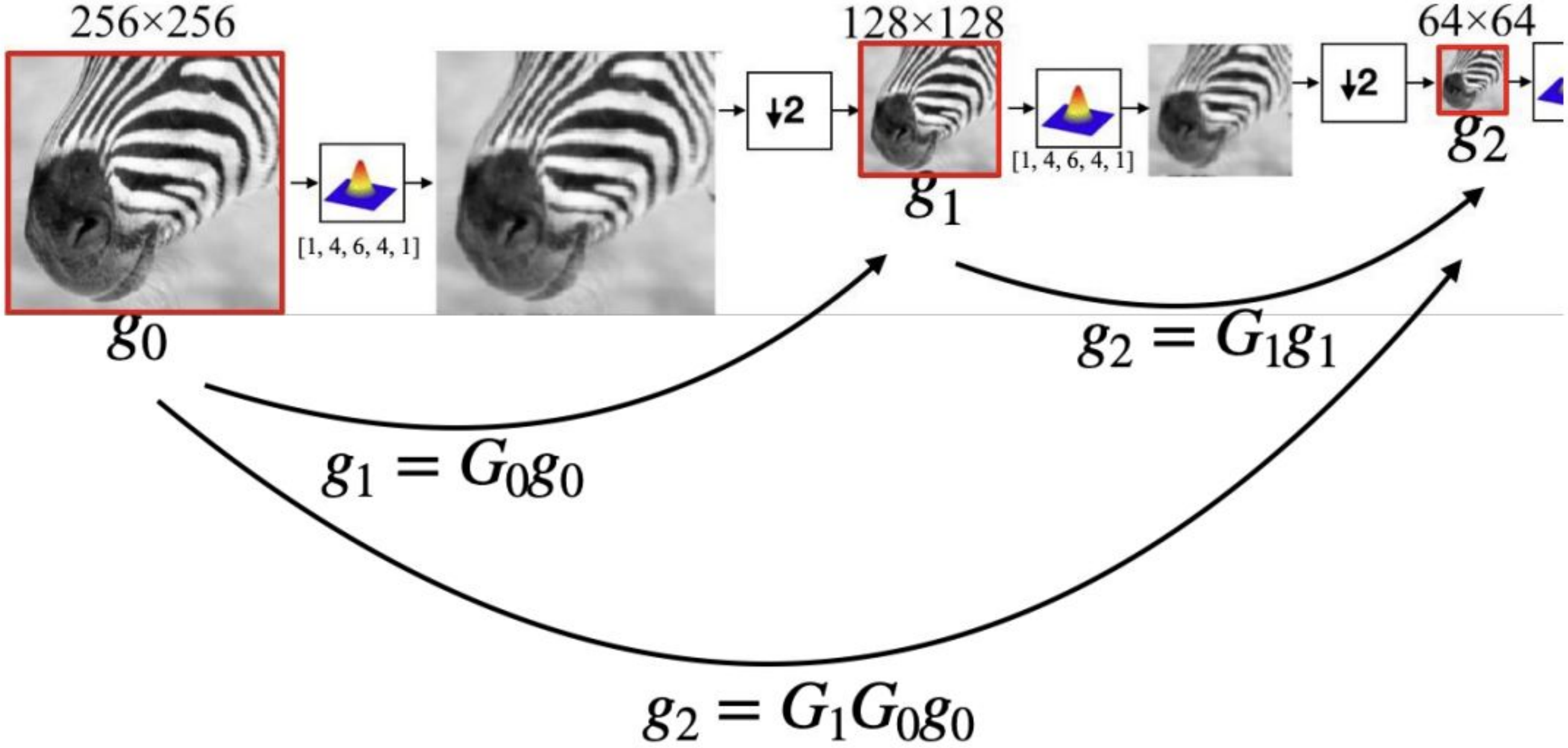


For each level

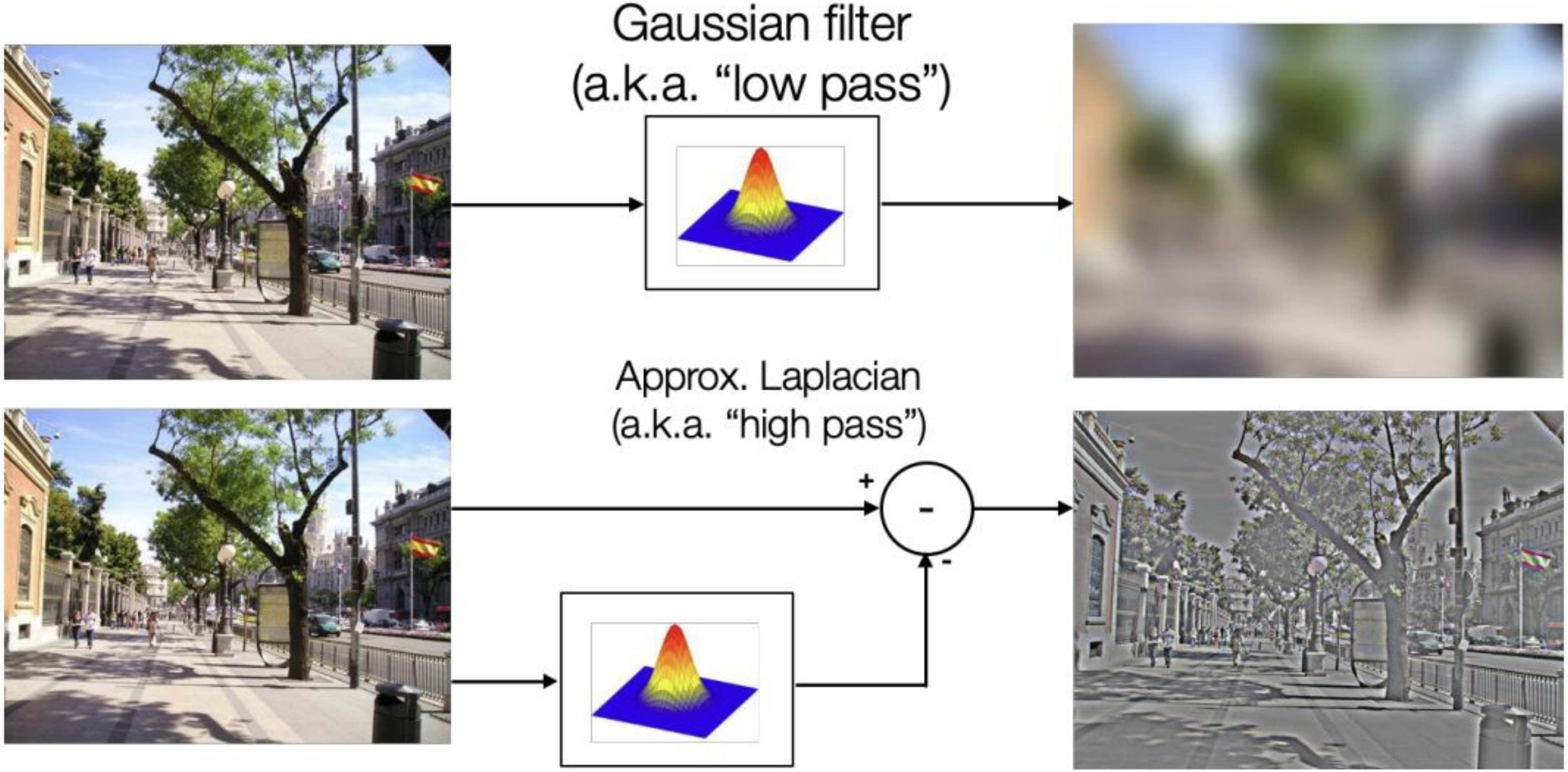
1. Blur input image with a Gaussian filter
2. Downsample image

What is Gk?
The operation including both blur and downsample

Gaussian Pyramid

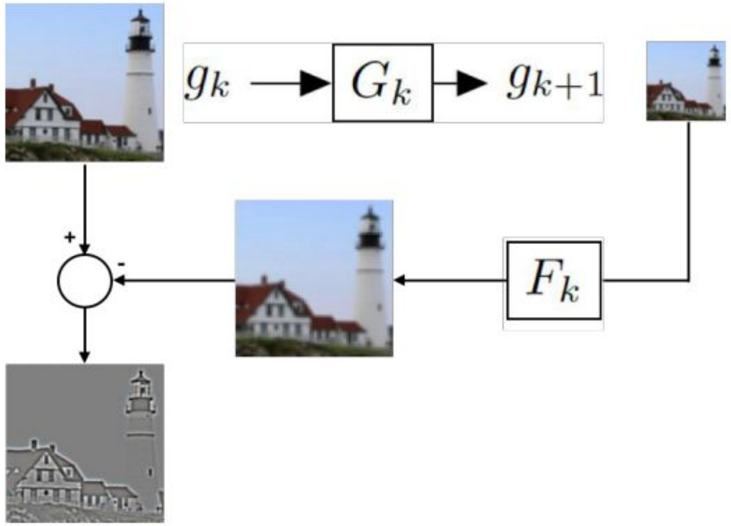


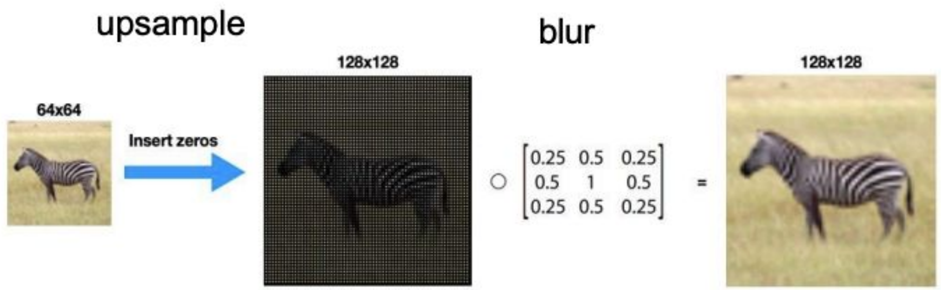
Use of Laplacian



Laplacian Pyramid

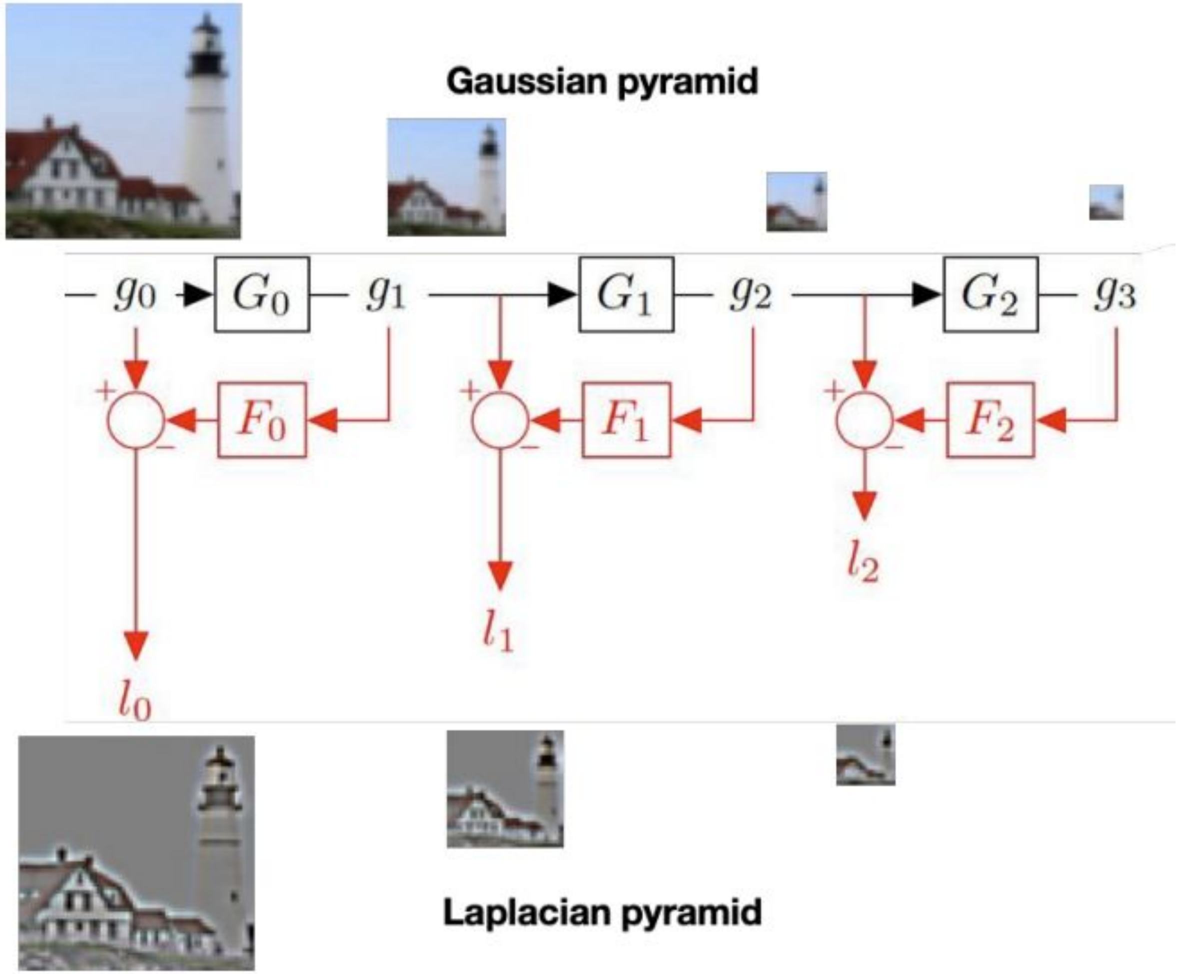
1. Upsample the Gaussian pyramid at level k+1
2. Blur the upsampled Gaussian pyramid at level k+1
3. The difference of Gaussian pyramid at level k and result from the 2nd step is the Laplacian pyramid



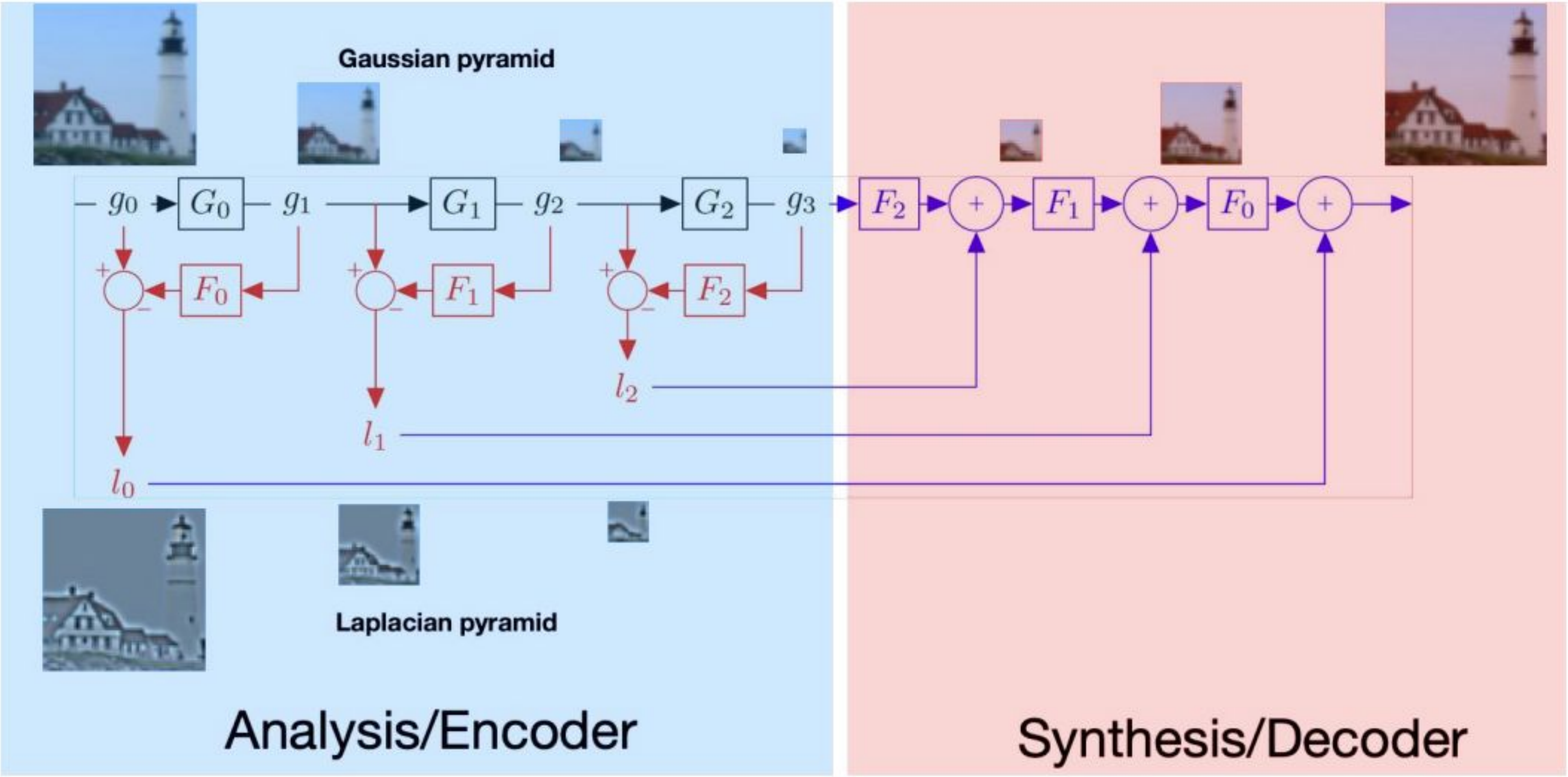


What is Fk?
The operation including both upsample and blur together

Laplacian Pyramid



Gaussian & Laplacian Pyramid



Gaussian & Laplacian Pyramid - Applications

- Texture synthesis
- Image compression
- Noise removal
- Computing image “kaypoints”

Image Blending (PS2)

- Build Laplacian pyramid for both images: L_A, L_B
- Build Gaussian pyramid for mask: G
- Build a combined Laplacian pyramid
- Collapse L to obtain the blended image