

Let's look at what the fourier transform of the Gaussian kernel results to:

$$\tilde{G}(k) := \mathcal{F}[\tilde{G}] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} e^{\pm i k x} dx \quad (1)$$

$$= \frac{1}{2\pi\sigma} e^{-\frac{\mu^2}{2\sigma^2}} \int_{-\infty}^{\infty} e^{-\frac{x^2 + (2\mu \pm i k 2\sigma^2)x}{2\sigma^2}} dx \quad (2)$$

$$= \frac{1}{2\pi\sigma} e^{-\frac{\mu^2}{2\sigma^2}} \int_{-\infty}^{\infty} e^{-\frac{[x - (\mu \pm i k \sigma^2)]^2 + (\mu \pm i k \sigma^2)^2}{2\sigma^2}} dx \quad (3)$$

$$= \frac{1}{2\pi\sigma} e^{-\frac{\mu^2}{2\sigma^2}} e^{\frac{\mu^2 \pm i \mu k \sigma^2 - k^2 \sigma^4}{2\sigma^2}} \int_{-\infty}^{\infty} e^{-\frac{[x - (\mu \pm i k \sigma^2)]^2}{\sigma^2}} dx \quad (4)$$

$$= \frac{1}{2\pi\sigma} e^{\pm i \mu k \sigma^2} e^{-\frac{k^2 \sigma^4}{2}} \sqrt{2\pi}\sigma \quad (5)$$

$$= \frac{1}{\sqrt{2\pi}} e^{\pm i \mu k \sigma^2} e^{-\frac{k^2 \sigma^2}{2}} \quad (6)$$

$$= \sigma e^{\pm i \mu k \sigma^2} G(\sigma x) \quad (7)$$

This can be done more abstract by using some Fourier transform rules. First we proof the fixpoint of the Fourier transform:

$$G(x) := \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \quad (8)$$

$$g(k) := \mathcal{F}[G] := \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} G(x) e^{i k x} dx = \frac{1}{\sqrt{2\pi}} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{i k x - \frac{x^2}{2}} dx \quad (9)$$

$$= \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-\frac{1}{2}(x - i k)^2 + i^2 \frac{k^2}{2}} dx = \frac{1}{2\pi} e^{-\frac{k^2}{2}} \int_{-\infty}^{\infty} e^{-\frac{1}{2}(x - i k)^2} dx \quad (10)$$

$$= \frac{1}{2\pi} e^{-\frac{k^2}{2}} \sqrt{2\pi} = \frac{1}{\sqrt{2\pi}} e^{-\frac{k^2}{2}} = G(k) \quad (11)$$

By expressing \tilde{G} using the fixpoint function G we can show how \tilde{G} transforms, Note that for brevity it looks like \mathcal{F} is applied on a function value instead of a function, but let's just say, that e.g. $\mathcal{F}[f(x)]$ actually means $\mathcal{F}[f]$. The mathematical way would be to define functionals (in physics operators) which

translate or scale the argument of a function, see also T_a below.

$$\tilde{G}(k) = \frac{1}{\sigma} G\left(\frac{x - \mu}{\sigma}\right) \quad (12)$$

$$\Rightarrow \mathcal{F}\left[\frac{1}{\sigma} G\left(\frac{x - \mu}{\sigma}\right)\right](x) \quad (13)$$

$$= \frac{1}{\sigma} \sigma \mathcal{F}[G(x - \mu)](\sigma k) \quad (14)$$

$$= e^{-i\mu k \sigma^2} \mathcal{F}[G](\sigma k) \quad (15)$$

$$= e^{-i\mu k \sigma^2} G(\sigma k) \quad (16)$$

$$= \frac{1}{\sqrt{2\pi}} e^{-i\mu k \sigma^2} e^{-\frac{k^2 \sigma^2}{2}} \quad (17)$$

Let's look what a periodic shift of the kernel will do to the convoluted result.
Let \mathbb{K} be \mathbb{C} or \mathbb{R} .

$$T_a : (\mathbb{K} \rightarrow \mathbb{K}) \rightarrow (\mathbb{K} \rightarrow \mathbb{K}), T_a[f](x) \mapsto f(x - a) \quad (18)$$

Then the translation rule for the Fourier transform can be written as:

$$\mathcal{F}[T_a[f]] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x - a) e^{-ikx} dx \quad (19)$$

$$\stackrel{y=x-a}{=} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(y) e^{-ik(y+a)} dx \quad (20)$$

$$= e^{-ika} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(y) e^{-iky} dx \quad (21)$$

$$= \frac{1}{\sqrt{2\pi}} e^{-ika} \mathcal{F}[f] \quad (22)$$

Here we define the function (not a functional) $\tilde{T}_a(k)$ which shifts in real space x when multiplied with a function in frequency-space k .

$$\tilde{T}_a(k) := \frac{1}{\sqrt{2\pi}} e^{-ika} \quad (23)$$

$$\mathcal{F}[\tilde{T}_a](x) = T_a[\delta](x) = \delta(x - a) \quad (24)$$

$$\mathcal{F}^{-1}[\tilde{T}_a](x) = T_{-a}[\delta](x) = \delta(x + a) \quad (25)$$

$$(26)$$

Function are to be equal if $\forall x : f(x) = g(x)$. With this we can take a closer look at shifted kernels G .

$$f * T_a[G] = \mathcal{F}^{-1}[\mathcal{F}[f] \cdot \mathcal{F}[T_a[G]]] \quad (27)$$

$$= \mathcal{F}^{-1}[\mathcal{F}[f] \cdot \tilde{T}_a \cdot \mathcal{F}[G]] \quad (28)$$

$$= \mathcal{F}^{-1}[\mathcal{F}[T_a[f]] \cdot \mathcal{F}[G]] \quad (29)$$

$$= T_a[f] * G \quad (30)$$

This means if we blur an image with a kernel which is shifted, then the resulting blurred image will be shifted, too. This means we need to create the Gaussian at $x = 0, y = 0$. For the FFTW-Library the 2D array is expected to lie in memory in row-major format, i.e. the coordinate (ix, iy) is at `data[iy*Nx+ix]`. This means the center of the Gaussian is at `data[0]` and the four nearest and therefore highest-value neighbors are at `data[1]`, `data[Nx]`, `data[(Ny-1)*Nx]`, `data[(Ny-1)*Nx+Nx-1]`. In this and only this case the Fourier transform of a real Gaussian kernel will also be a real Gaussian, as seen above. This can be used by checking if the transformed kernel has no imaginary part.

ToDo:

- Be more rigorous about 2π factors in convolution theorem
- Relate results for Fourier Transform to the actually used Discrete Fourier Transform https://en.wikipedia.org/wiki/Discrete_Fourier_transform#Circular_convolution_theorem_and_cross-correlation_theorem Basically everything we used, like the translation or convolution theorem also exist for the DFT, therefore the proofs above also hold true for the DFT.

Initially the kernel and the input image in Figure 1 are black and white images with pixels in $[0.0, 1.0]$, but after the first transform they are always complex valued. The complex values are visualized with the magnitude as luminosity and the phase being converted to a hue. A zero phase, i.e. a real value, maps to a red hue. Note that perfect white can't be colored with the hue in the HSL-scheme, therefore the phase information can't be plotted inside the triangle. It can be seen, that the resulting blurred image has some very small non-zero phases, because of numerical errors.

As can be seen, because the Gaussian kernel is not perfect at $(x, y) = (0, 0)$, but at $(50, 50)$ the triangle gets shifted. This means the kernel must be at zero, which is the upper left corner which corresponds to the very first value in the linearized 1D array of complex values as FFTW or cuFFT uses it.

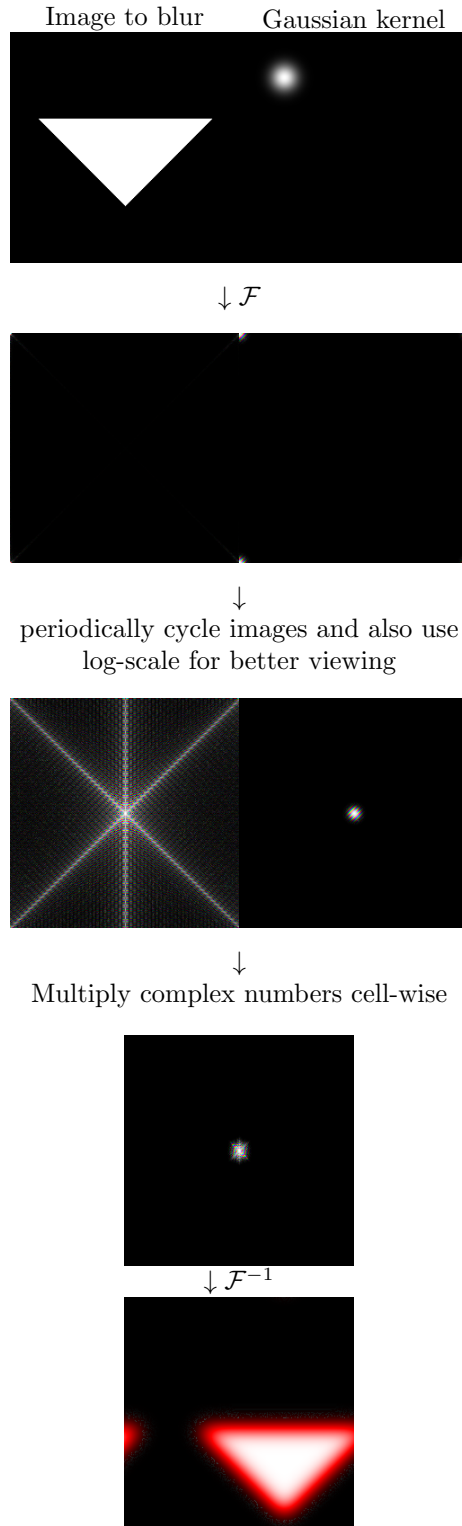


Figure 1: Exemplary blurring of a 256×256 large image containing a triangle with a Gaussian with $\sigma = 10$ at position $(50, 50)$.