

→ FFT reduces complexity from n^2 to $n \log n$.

Discrete Fourier Transform in 2D

1D \rightarrow k in $N \times X$

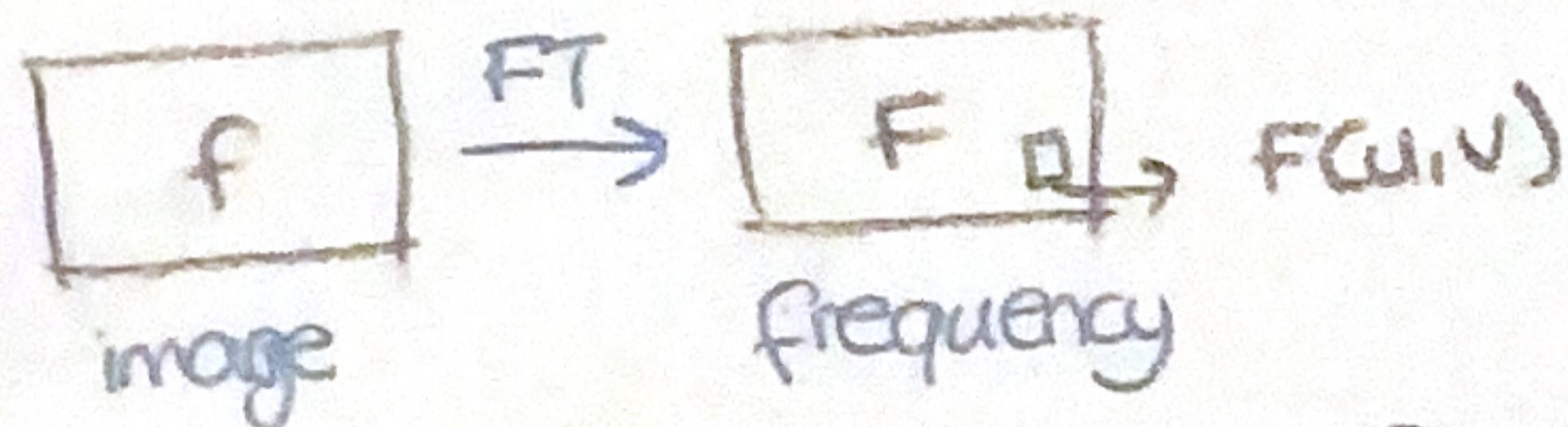
2D \rightarrow uv xy MN f F

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-i 2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right)}$$

↓
pixel of the
frequency transformed
image

→ u & v are the frequencies

→ A frequency in an image corresponds to change in an image



How to calculate Fourier in 2D?

def myDFT2D(f)

for u=0:M-1

 for v=0:N-1

 for x=0:M-1

 for y=0:N-1

 output[u+1, v+1] = output[u+1, v+1] +
 output[u+1, v+1] = output[u+1, v+1] +
 f[x+1, y+1] * exp(-2*pi*i*(x+u/M +
 y+v/N))