Image Analysis and Processing

Transforms

Laurent Najman

laurent.najman@esiee.fr

ESIEE Paris

Université Paris-Est, Laboratoire d'Informatique Gaspard-Monge, Équipe A3SI

Before we start

- How are we doing?
- Questions?
- Concerns?
- Tutorial problems?
    - Programming problems?

Syllabus

1. Introduction, image perception and representation
2. Enhancements – Histogram & pixelwise transforms.
3. **Transforms – FFT, Laplace, Z, Hough.**
4. Filtering – Linear filters.
5. Segmentation I
6. Segmentation II
7. Applications

What are transforms?

Let I be an image represented as a function, $I : \mathbb{R}^n \longrightarrow \mathbb{R}$,
Then $\mathcal{T}$, a transform, is simply an operator on I:

$$J = \mathcal{T}(\mathcal{I})$$

In practice however most operators are not called "transforms" ; this term is derived from "integral transforms" of which the FOURIER and LAPLACE transforms are parts of.
By extension, transforms are those that define broad classes of operators, and/or which allows for a different *representation* of the same data using different bases.

Example of transforms relevant to IP

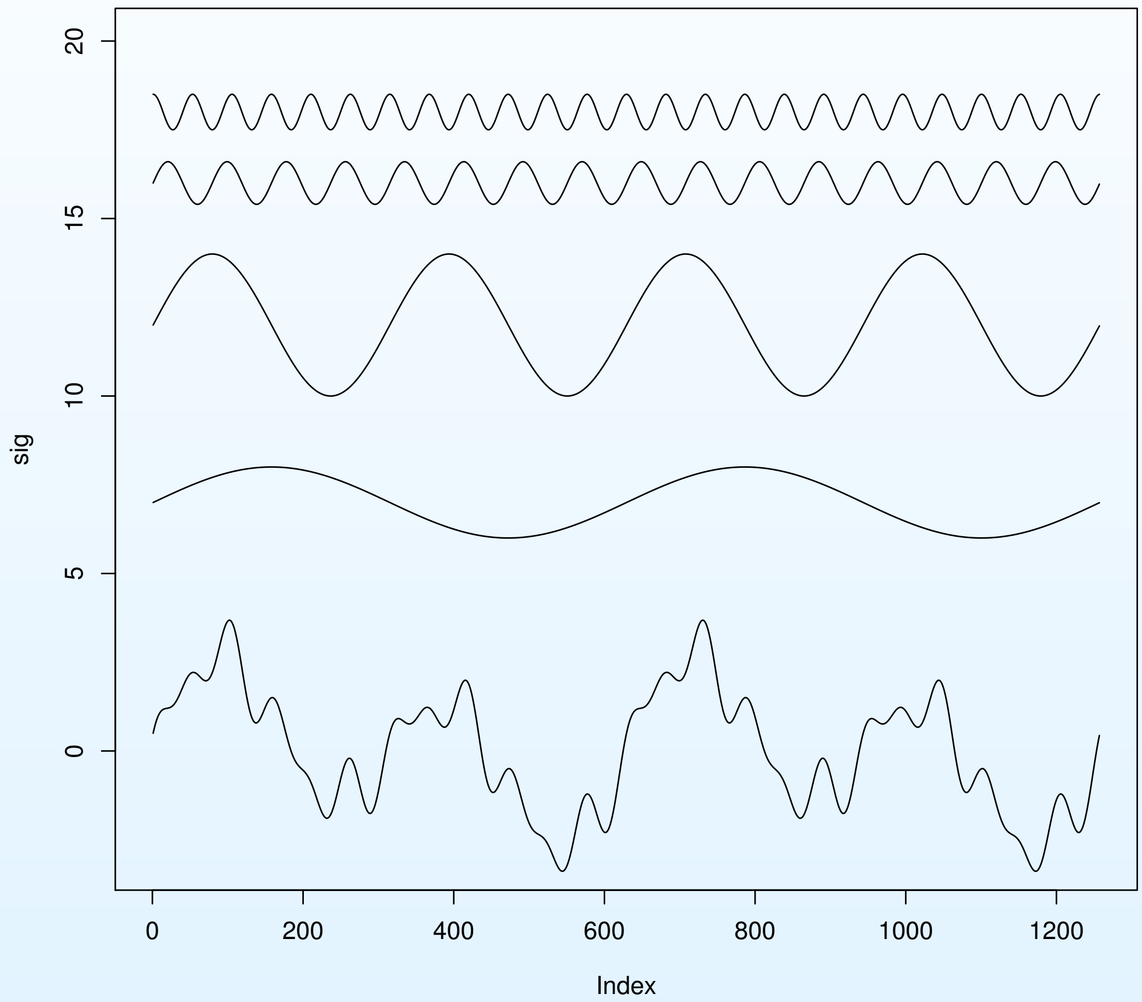
- FOURIER transform and its derivatives: DFT, FFT.
- Discrete cosine transform (DCT): used in coding (JPEG).
- KARHUNEN-LOÈVE transform (KLT): optimal coding.
- LAPLACE and Z transforms: exponential filters (1D).
- HOUGH transform: line and curve detection.
- Wavelet transforms: coding, filtering, texture representation.

The FOURIER *transform*

Background

- Named after Jean Baptiste Joseph Fourier (b. 1768)
- 1807: memoir, 1822: Book: “Théorie Analytique de la Chaleur”.
- Translated 1878 (Freeman) “Analytic theory of heat”.
- Idea that periodic signals could be decomposed as series of sines and cosines (Fourier series)
- Idea that non-periodic but finite-area functions can also be represented as an integral sum of sines and cosines: the Fourier transform.
- Practical and useful idea that took more than 100 years to be “digested”.
- Really took off with the advent of computers and the FFT 50 years ago.

Periodic signals as sum of sines



`sin(x) + 2 * sin(2*x)+ 0.6*sin(8*x) + 0.5 * cos(12*x)`

Continuous FT

- The FOURIER transform is a prime example of *integral* transform:
- Forward transform:

$$F(u) = \int_{-\infty}^{+\infty} f(x) e^{-j2\pi ux} dx$$

- Inverse transform: $f(x) = \int_{-\infty}^{+\infty} F(u) e^{j2\pi ux} du$
- FOURIER pairs: $f(x) \Leftrightarrow F(u)$
- x and u complex.
- Existence subject to: $\int_{-\infty}^{+\infty} |f(x)| dx$ exists and is finite, f has a finite number of discontinuities, f has bounded variations.

2D FOURIER transform

Let $f(x,y)$ be a 2D image, a function $I\!R^2 \longrightarrow I\!R$. Its FOURIER transform can be derived in the following fashion (separability):

$$\begin{aligned} F_1(u,y) &= \int_{-\infty}^{+\infty} f(x,y)e^{-j2\pi xu}dx \\ F(u,v) &= \int_{-\infty}^{+\infty} F_1(u,y)e^{-j2\pi yv}dy \\ &= \int_{-\infty}^{+\infty}\left[\int_{-\infty}^{+\infty} f(x,y)e^{-j2\pi xu}dx\right]e^{-j2\pi yv}dy \\ &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)e^{-j2\pi(xu+yv)}dxdy \end{aligned}$$

Similarly: $f(x,y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} F(u,v)e^{j2\pi(xu+yv)}dxdy$

Discrete FOURIER transform

Let f be a discrete function $[0, M[\subset \mathbb{Z} \longrightarrow I\!R$, then its DFT is given by

$$F(u) = \sum_{x=0}^{M-1} f(x) e^{-j(2\pi xu)/M}$$

Similarly as before, the inverse DFT is given by:

$$f(x) = \sum_{x=0}^{M-1} F(u) e^{j(2\pi xu)/M}$$

- Q1: how many operations to compute the DFT (as a function of M)?
- Q2: is existence a problem?
- Q3: 2D versions?

Warning: be alert (not alarmed)

- We'll use the continuous FOURIER transform (CFT) or the DFT somewhat interchangably.
- We'll show proofs on the CFT if convenient.
- We'll use the 1-D DFT for basic properties, moving to 2-D and more later.
- We'll try to repeat things in different contexts (1-D, 2-D, continuous/discrete).

Frequency domain

- Other way to write the DFT:

$$F(u) = \sum_{x=0}^{M-1} f(x)[cos(2\pi ux/M) - jsin(2\pi ux/M)]$$

- Each term of F is the sum of all values of f weighted by sines and cosines of various frequencies. F is the *frequency domain* representation of f.
- Polar representation:

$$F(u) = \|F(u)\|e^{-j\phi(u)};$$

$$\|F(u)\| = \left[\mathrm{Re}(F(u))^2 + \mathrm{Im}(F(u))^2\right]^{1/2}, \phi(u) = \tan^{-1}\left[\frac{\mathrm{Im}(F(u))}{\mathrm{Re}(F(u))}\right].$$

Note on sampling

- We sample $f(x)$ at $x = 0, 1, 2, ...$
- These are *not* necessarily integer samples. The sampling is uniform of width Δx but arbitrary, we mean:

$$f(x) = f(x_0 + x * \Delta x)$$

- Similarly, $F(u)$ is also sampled, but always starts at zero, i.e:

$$F(u) = F(u\Delta u)$$

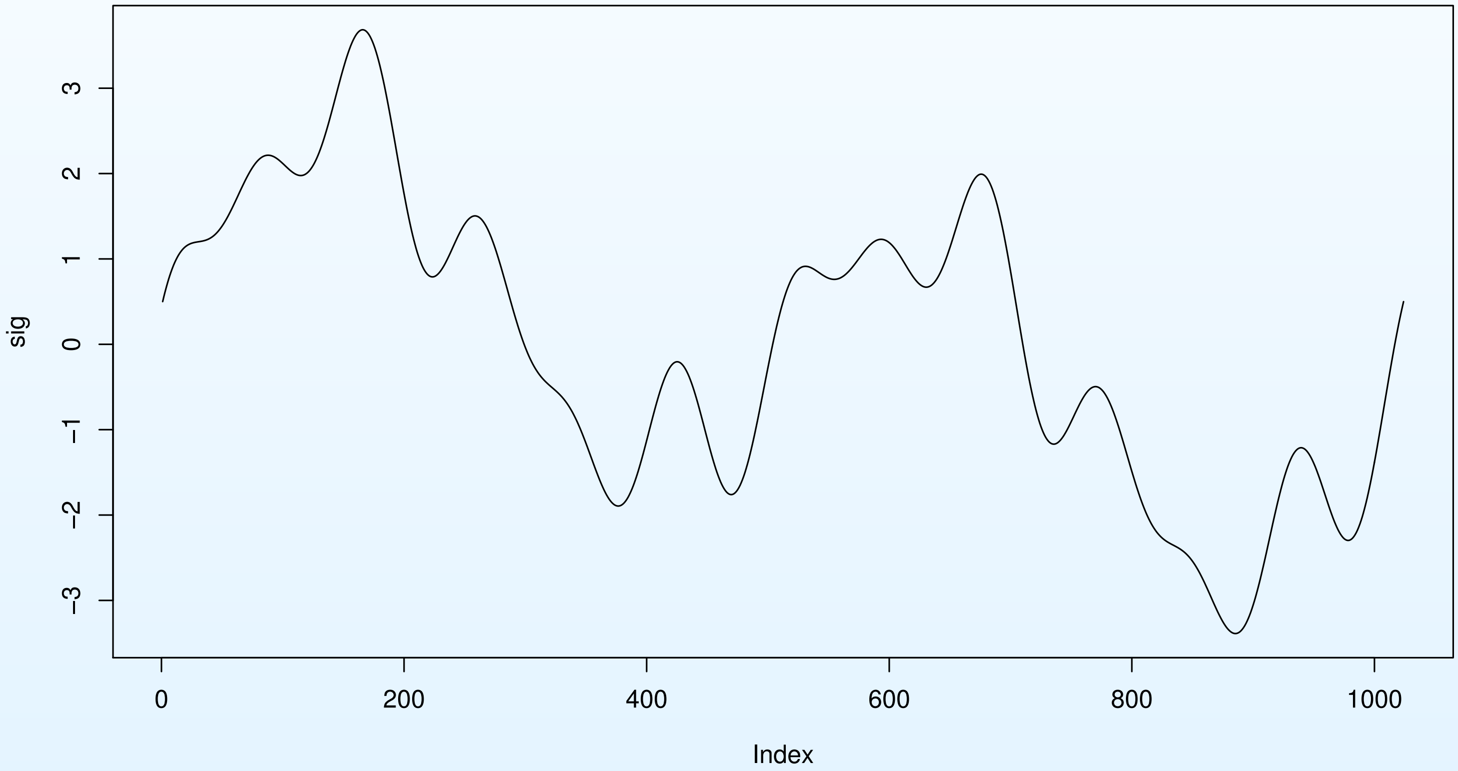
- We have the following relationship between samplings:

$$\Delta u = \frac{1}{M\Delta x}$$

- The DFT does not have infinite domain: assumption of periodicity.

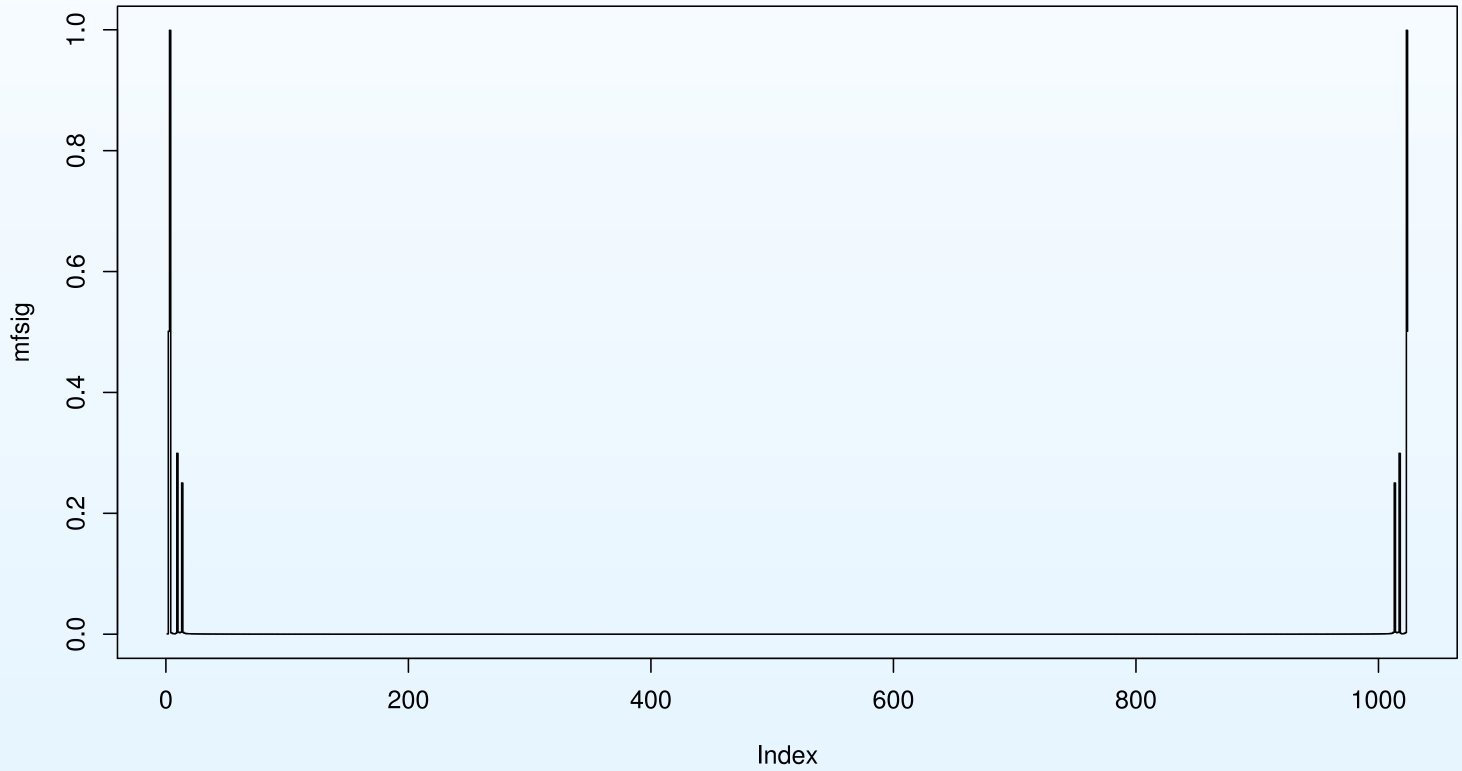
Example on periodic signal

```
x <- seq(0,2*pi,length=1024)
s <- sin(x) + 2 * sin(2*x)+ 0.6*sin(8*x) + 0.5 * cos(12
```



DFT output analysis

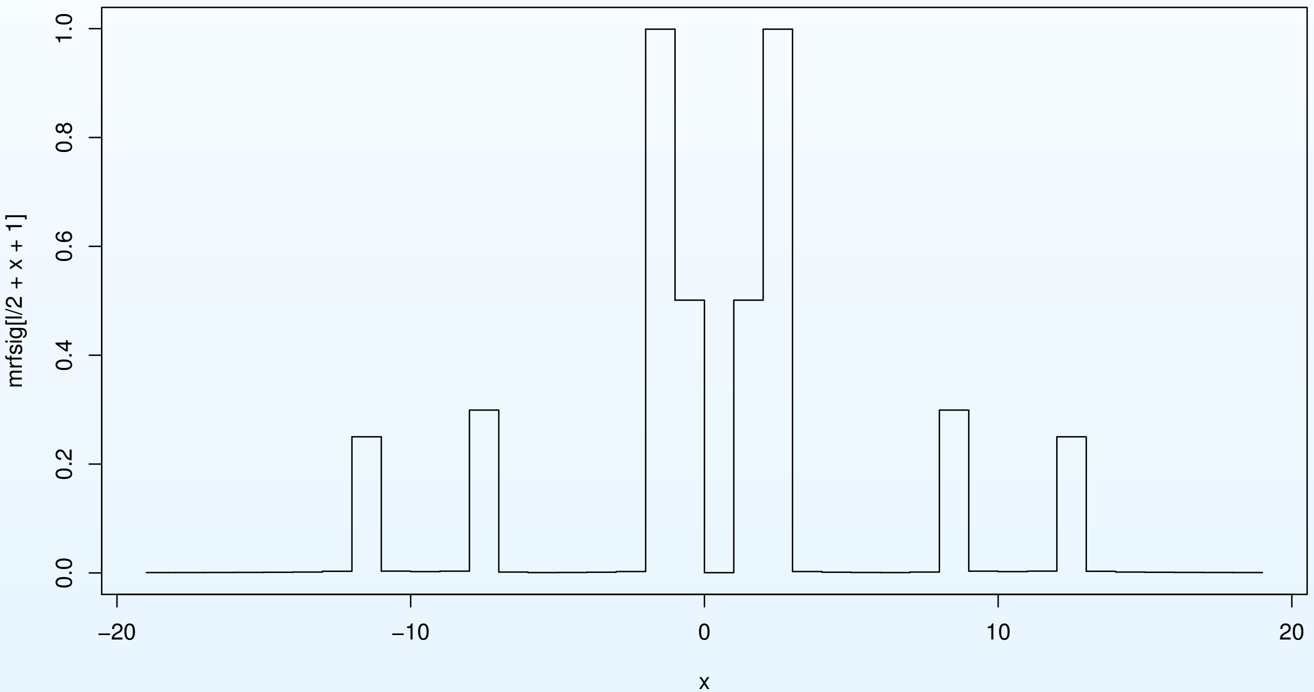
Raw DFT Output of preceding signal looks likes this:



Need to recenter the signal, only plot useful bits:

Analysis of periodic signal:

Signal: `s = sin(x) + 2 * sin(2*x)+ 0.6*sin(8*x) + 0.5 * cos(12*x)`

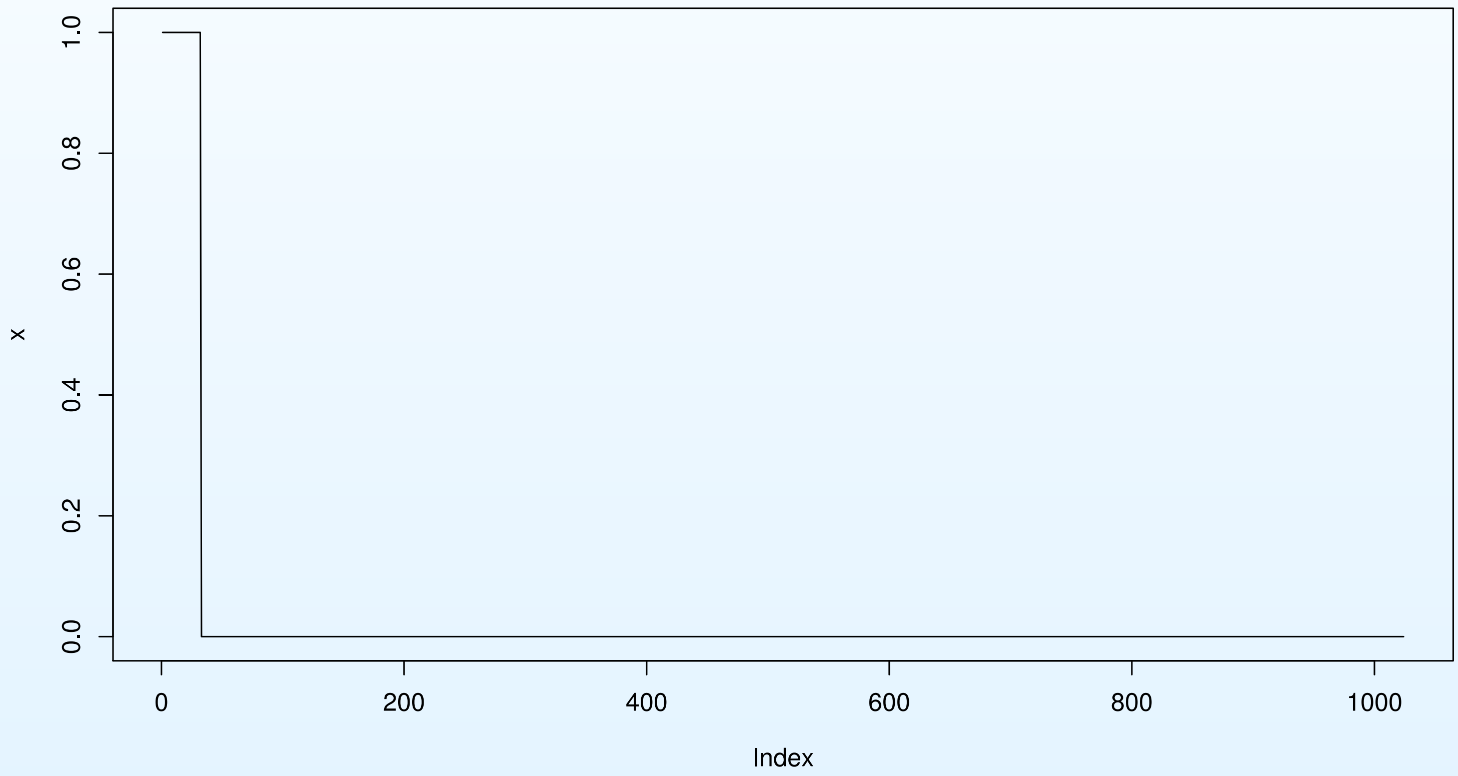


$$\mathrm{dft}(s)[1] = (0,0), \mathrm{dft}(s)[2] = (0.5, -\pi/2),$$

$$\mathrm{dft}(s)[3] = (1, -\pi/2), \mathrm{dft}(s)[9] = (0.3, -\pi/2), \mathrm{dft}(s)[13] = (0.25, 0)$$

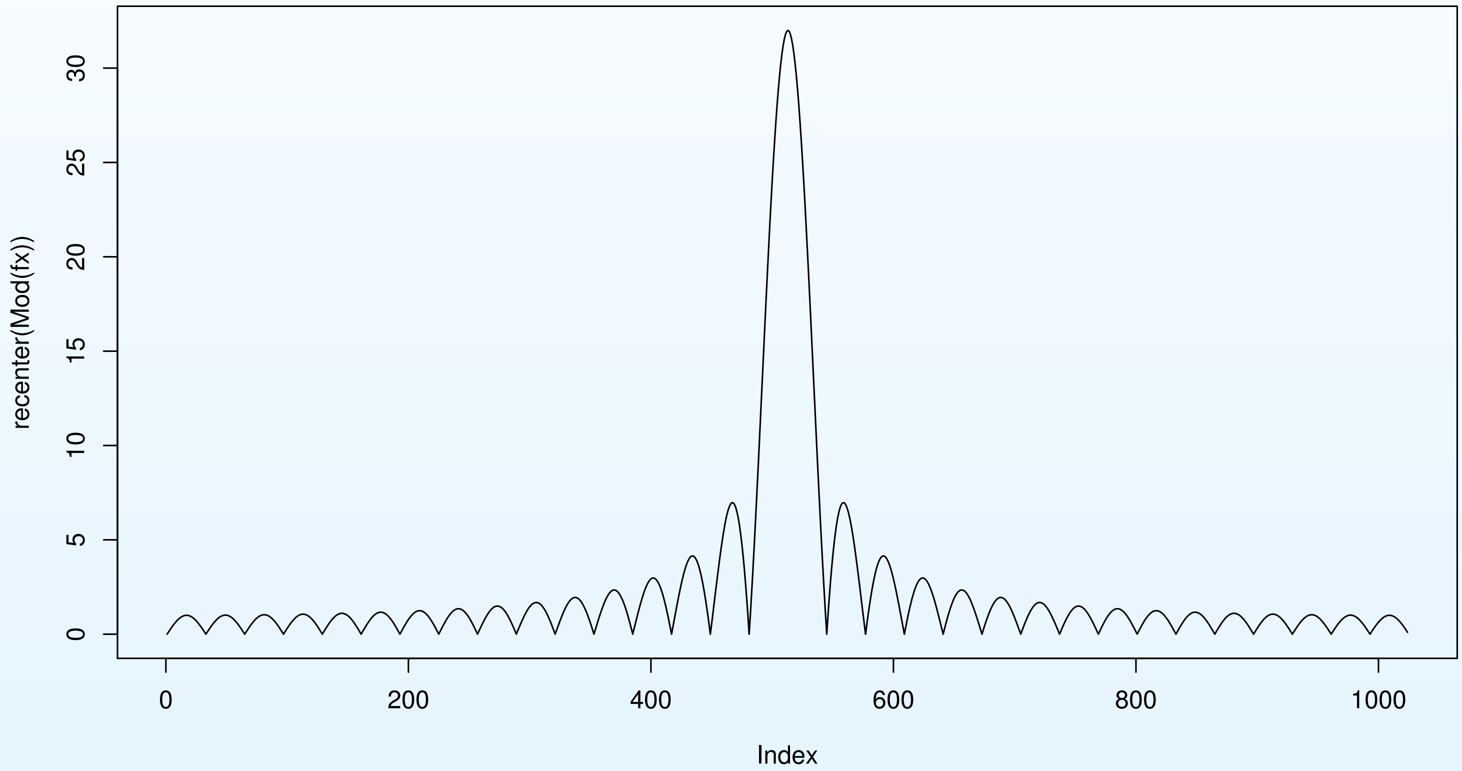
Non-periodic example: step

```
x <- seq(0,0,length=1024)
x[0:32] <- 1
```



Non-periodic example

DFT of a step:



Close form CFT of the box function

Let f be the box function:

$$\begin{aligned} f(x) &= 1 \text{ if } 1 \geq x \geq -1 \\ f(x) &= 0 \text{ otherwise} \end{aligned}$$

$$\begin{aligned} F(f(x))(u) &= \int_{-\infty}^{+\infty} f(x) e^{-j2\pi x u} dx \\ &= \int_{-1}^{+1} e^{-j2\pi x u} dx \\ &= \frac{1}{-j2\pi u} \left[e^{-j2\pi x u} \right]_{-1}^{+1} \\ &= \frac{\sin 2\pi u}{\pi u} \end{aligned}$$

Properties of the FOURIER transform

- Linearity: $F(a.f + b.g) = a.F(f) + b.F(g)$ (from linearity of the integral)
- Translation invariance: $f(x - x_0) \Leftrightarrow e^{-j2\pi x_0 u} F(u)$.
  note that $\|F(f(x - x_0))\| = \|F(f(x))\|$.
- Conversely $F(u - u_0) \Leftrightarrow f(x) e^{j2\pi u_0 x}$. Useful for recentering the DFT.
  - in discrete form: $f(x) e^{j2\pi x u_0 / M} \Leftrightarrow F(u - u_0)$
  - if $u_0 = M/2$, $e^{j2\pi x u_0 / M} = e^{j\pi x} = (-1)^x$
  - then, $f(x)(-1)^x \Leftrightarrow F(u - M/2)$.
- FOURIER transform of the derivative: $F(\frac{d^n f}{dx^n}) = (2\pi j u)^n F(f)$
- Derivative of the FOURIER transform:

$$\frac{d^n F(f(x))(u)}{du^n} = (-2\pi j x)^n F(f(x))(u)$$

Proof of the derivative property

$$
\begin{aligned}
F(\frac{df}{dx})(u) &= \int_{-\infty}^{+\infty} \frac{df(x)}{dx} e^{-j2\pi xu} dx \\
\int a^{'} b &= [ab] - \int ab^{'} \text{ (integration by part)} \\
&= \left[f e^{-j2\pi xu} \right]_{x=-\infty}^{x=+\infty} - \int_{-\infty}^{+\infty} f(x)(-j2\pi u) e^{-j2\pi ux} dx \\
&= 0 + j2\pi u \int_{-\infty}^{+\infty} f e^{-j2\pi ux} dx \\
&= j2\pi u F(f)(u)
\end{aligned}
$$

Note that: $\int_{-\infty}^{+\infty} |f| dx < +\infty \Rightarrow f(-\infty) = f(+\infty) = 0$

Repeat the process to get the final result: $F(\frac{d^n f}{dx^n}) = (2\pi j u)^n F(f)$.

2-D DFT

- Forward 2-D DFT

$$F(u,v) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(ux/M+yv/N)}$$

- Inverse 2-D DFT

$$f(x,y) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} F(u,v)e^{j2\pi(ux/M+vy/N)}$$

- Polar version

$$\begin{aligned} \|F(u,v)\| &= \left[\mathrm{Re}(F(u,v))^2 + \mathrm{Im}(F(u,v))^2\right]^{1/2} \\ \phi(u,v) &= \tan^{-1}\left[\frac{\mathrm{Im}(F(u,v))}{\mathrm{Re}(F(u,v))}\right] \end{aligned}$$

Properties of the 2-D DFT

- DC component:

$$F(0,0) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)$$

- Symmetry:

$$F(u,v) = F^*(-u,-v), \text{ therefore } \|F(u,v)\| = \|F(-u,-v)\|$$

- Sampling: $\Delta u = \frac{1}{M\Delta x}, \Delta v = \frac{1}{N\Delta y}$
- Re-centering: $f(x,y)e^{j2\pi(u_0 x/M + v_0 y/N)} \Leftrightarrow F(u-u_0, v-v_0)$ (Q:how?)
- Translation invariance:
  $f(x-x_0, y-y_0) \Leftrightarrow F(u,v)e^{-j2\pi(\frac{ux_0}{M}+\frac{vy_0}{N})}$ (Q:why?)

Properties of the 2-D DFT (2)

- Linear (as in the 1-D case)
  - Scaling: $F(a.f) = a.F(f)$
  - Distributivity: $F(f+g) = F(f) + F(g)$
  - $f(ax, by) \Leftrightarrow \frac{1}{|ab|} F(u/a, v/b)$
  - *Note:* $F(f.g) \neq F(f).F(g)$
- Rotation in spatial and frequency domain linked. Let:

$$x = r\cos\theta, y = r\sin\theta, u = \omega\cos\phi, v = \omega\sin\phi$$

Then $f(x, y)$ and $F(u, v)$ become $f(r, \theta)$ and $F(\omega, \phi)$ respectively, and:

$$f(r, \theta + \theta_0) \Leftrightarrow F(\omega, \phi + \theta_0)$$

(using the FOURIER transform in Polar coordinates).

Properties of the 2-D DFT (3)

- Periodicity:

$$F(u,v) = F(u+M,v) = F(u,v+N) = F(u+M,v+N)$$

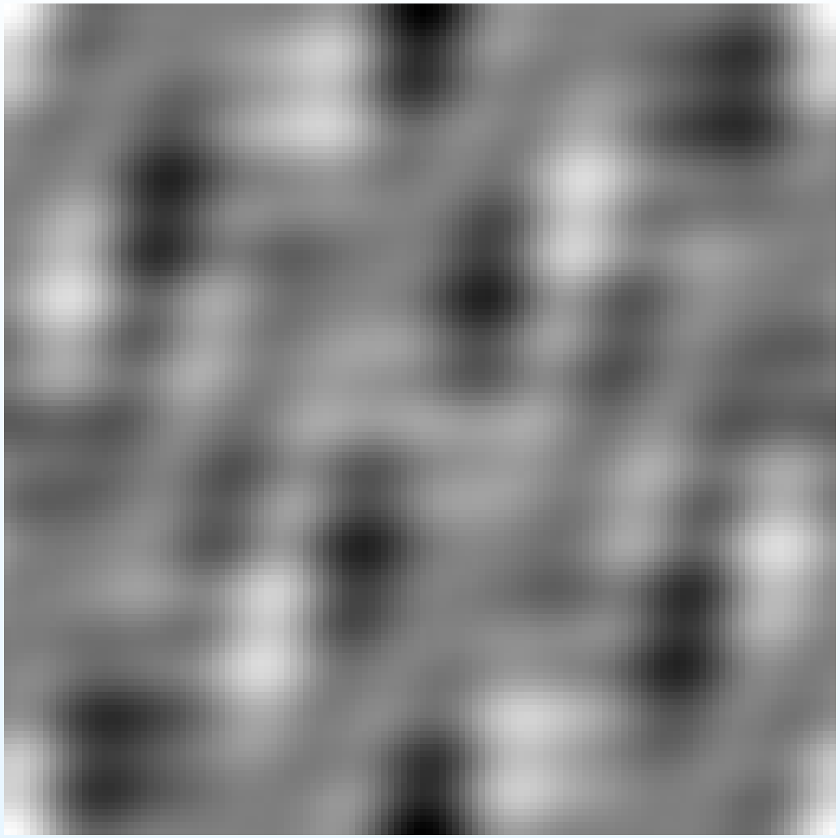
$$f(x,y) = f(x+M,y) = f(x,y+N) = f(x+M,y+N)$$

- Spectrum centered on origin (need to recenter):

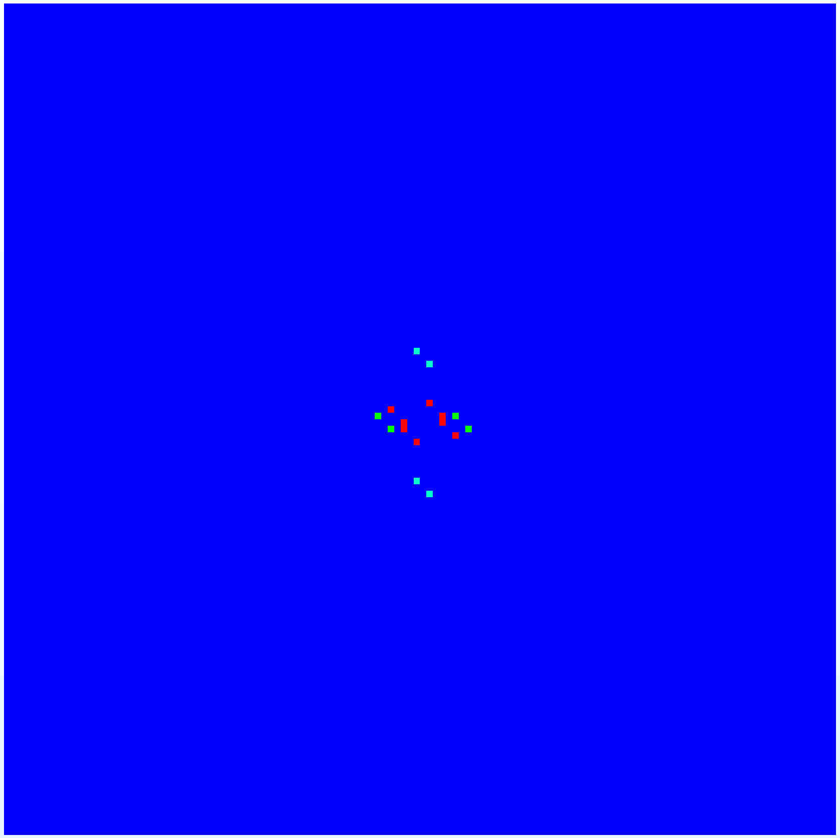
$$F(u,v) = F^*(-u,-v) \Longrightarrow |F(u,v)| = |F(-u,-v)|$$

  To re-center, multiply $f(x,y)$ by $(-1)^{x+y}$.
- Separability (from the continuous definition).

Example periodic image



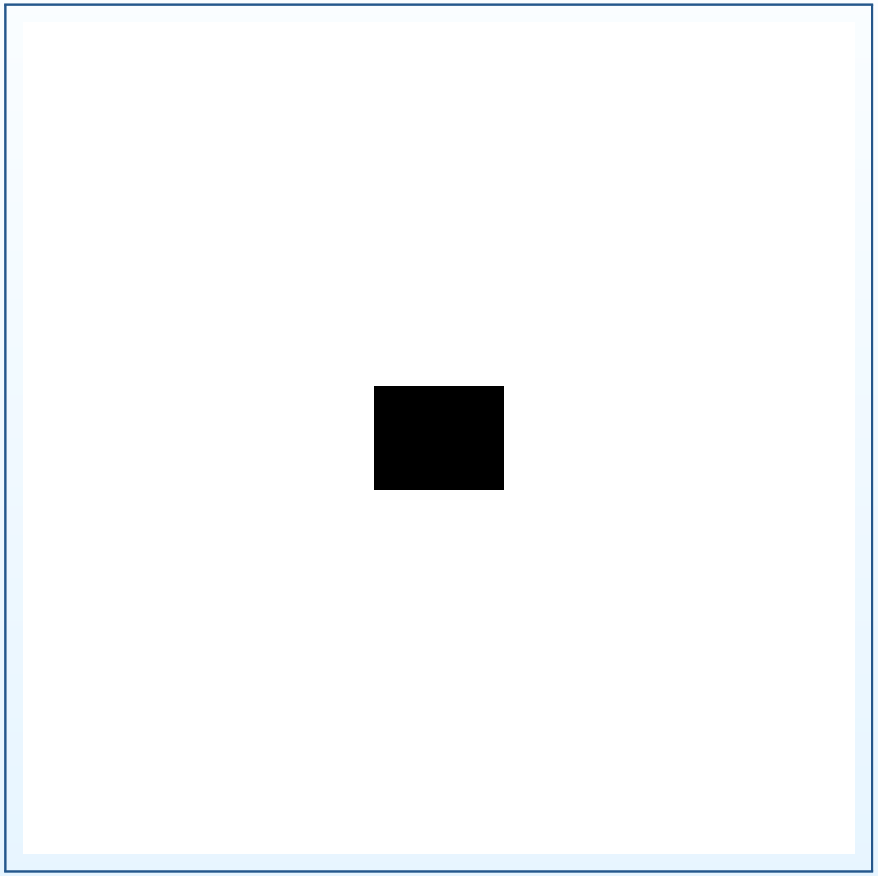
Synthetic texture image



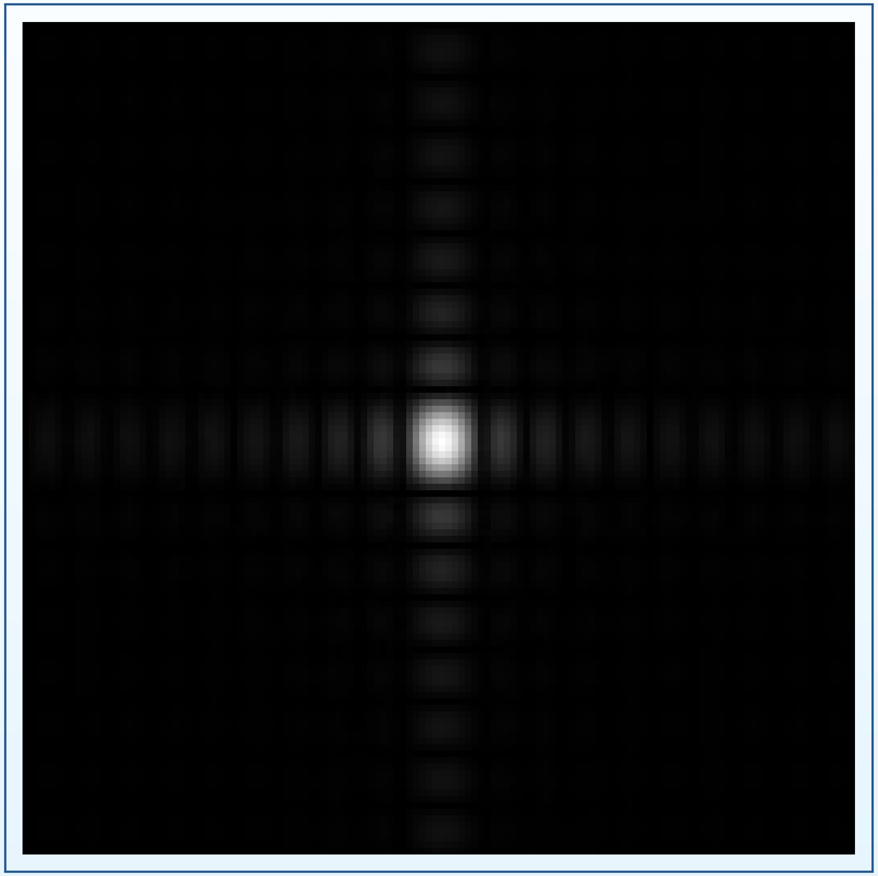
Modulus of its DFT

Image synthesized by setting random peaks (in symmetric pairs) in an empty image, then doing an inverse DFT.

Example non-periodic image



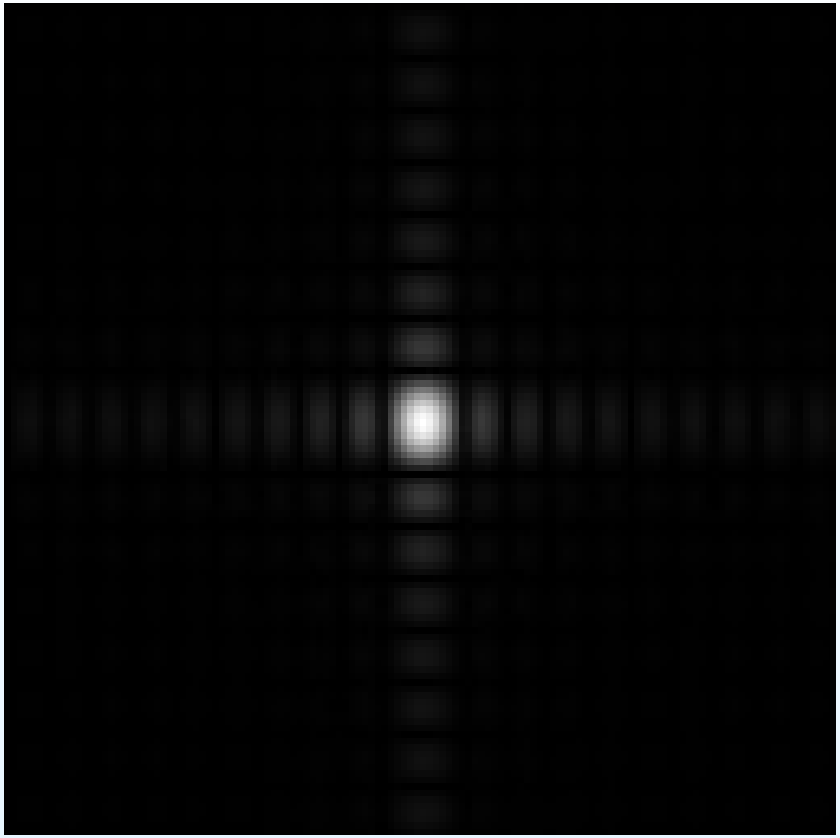
Box image



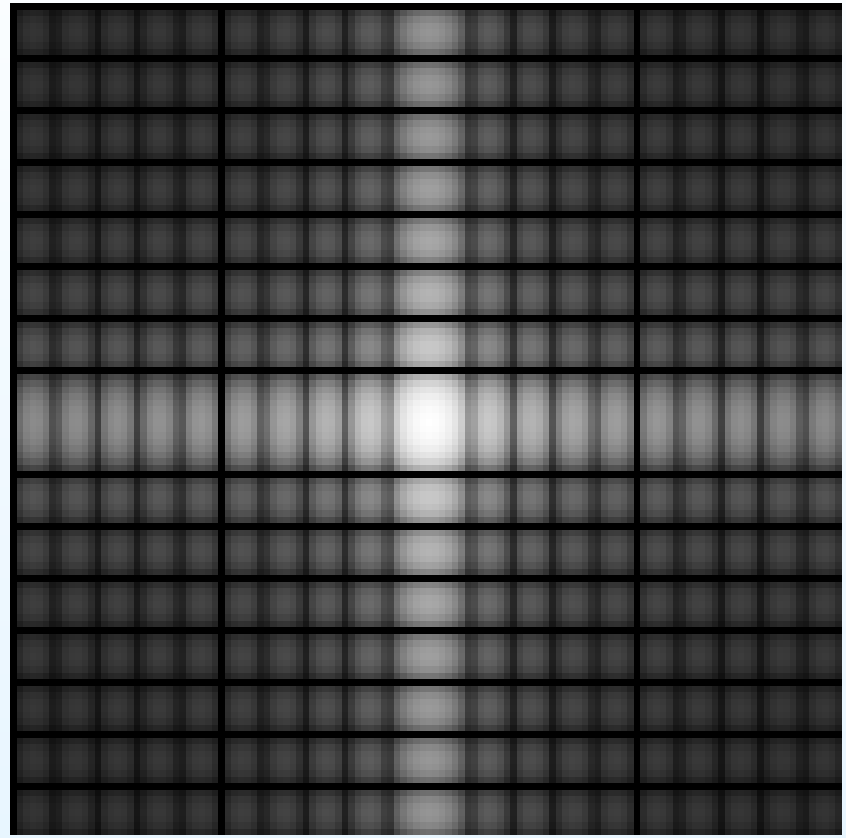
Modulus of its DFT

Corresponding image to the step function in the 1-D case.

Contrast problem with DFT images



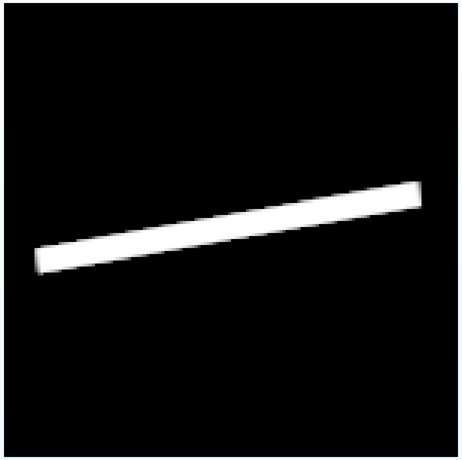
DFT image



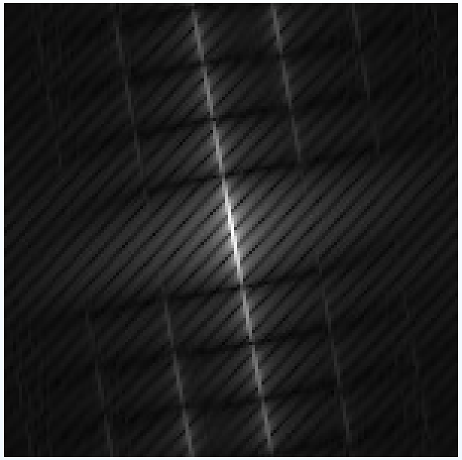
After log transform

If D is the image on the left, the image on the right is $\log(D+1)$. This reduces excessive contrast while keeping zeroes intact.

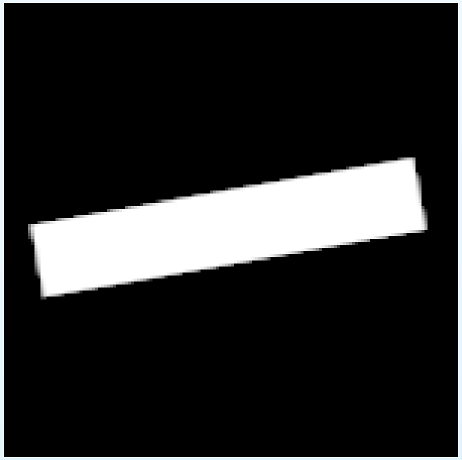
DFT of an edge



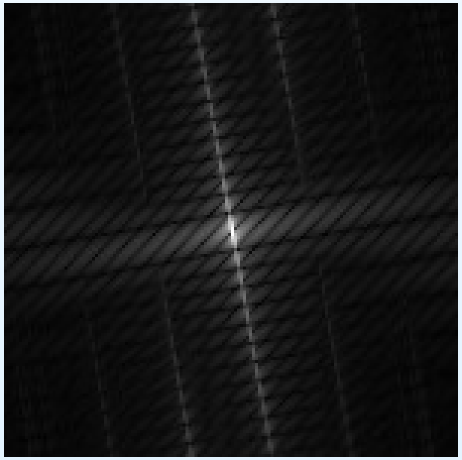
Thin edge



Thin edge DFT

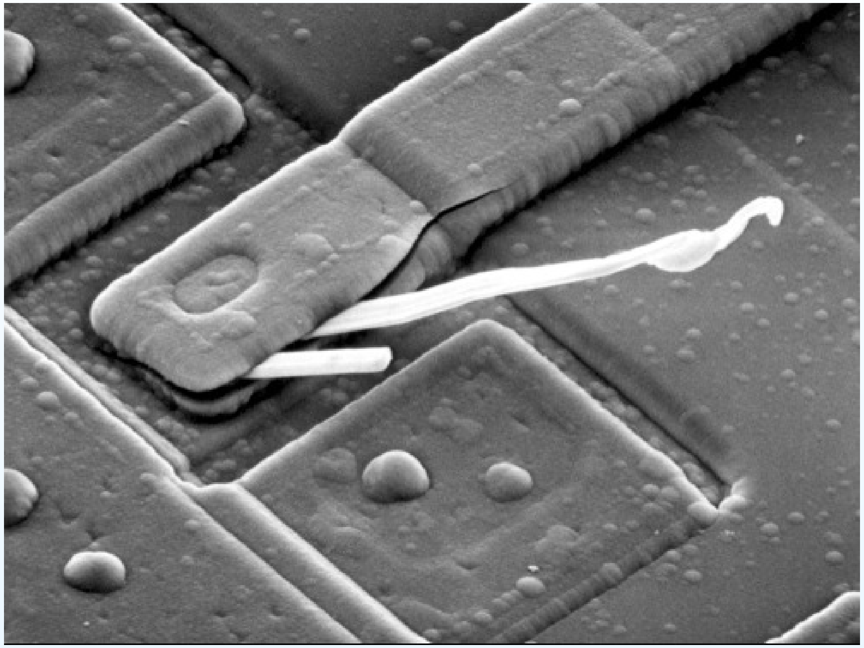


Thick edge

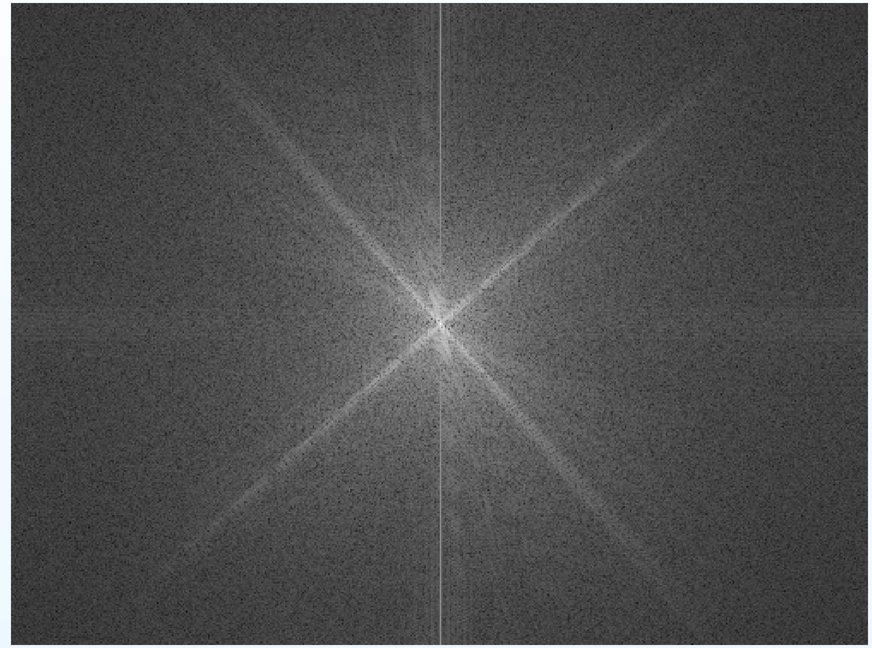


Thick edge DFT

DFT of a real image



SEM micrograph



its DFT

Notice: strongs edges at 45^o, extrusions, and corresponding features in the DFT.

Filtering in the frequency domain

Convolutions

- Continuous convolution (1-D):

$$(f * g)(x) = \int_{-\infty}^{+\infty} f(h)g(x-h)dh$$

  Here is an animation, here is another one
- in 2-D discrete, f and g both of size $M \times N$:

$$f(x,y) * g(x,y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n)g(x-m, y-n)$$

- *Equivalent* to weighted moving average.
- Q: in the spatial domain, how many operations are needed for convolving two $N \times N$ images?

Convolution and FOURIER transform

- Convolution theorem: Let f, g be a function pair and F, G their FOURIER transforms, then:

$$f(x,y) * g(x,y) \Leftrightarrow F(u,v).G(u,v)$$

  and

$$f(x,y).g(x,y) \Leftrightarrow F(u,v) * G(u,v)$$

  Where . is the element-by-element standard multiplication.

- if

$$f(x,y) * h(x,y) \Leftrightarrow F(u,v) * H(u,v)$$

  Then

$$h(x,y) \Leftrightarrow H(u,v)$$

  A filter designed in frequency domain yields a filter in the spatial domain, and vice-versa.

Proof of the convolution theorem (1-D)

$$\begin{aligned}
\mathcal{F}_u((f*g)(x))(u) &= \int_{-\infty}^{+\infty} (f*g)(x)e^{-j2\pi xu}dx \\
&= \int_{-\infty}^{+\infty} \left[\int_{-\infty}^{+\infty} f(h)g(x-h)dh\right] e^{-j2\pi xu}dx \\
&= \int_{-\infty}^{+\infty} f(h) \left[\int_{-\infty}^{+\infty} g(x-h)e^{-j2\pi xu}dx\right] dh \\
&= \int_{-\infty}^{+\infty} f(h) \left[\int_{-\infty}^{+\infty} g(x-h)e^{-j2\pi(x-h)u}dx\right] e^{-j2\pi hu}dh \\
&= \int_{-\infty}^{+\infty} f(h)G(u)e^{-j2\pi hu}dh \\
&= G(u).\int_{-\infty}^{+\infty} f(h)e^{-j2\pi hu}dh \\
&= G(u).F(u)
\end{aligned}$$

Steps for filtering using the DFT

1. Forward DFT of the input image;
2. Recenter;
3. Design of filter;
4. Padding to avoid edge effects;
5. Product of DFT and filter. If filter is real, leaves the phase intact;
6. Decenter;
7. Inverse DFT;
8. Remove padding.

Q: in the frequency domain, how many operations are needed for convolving two $N \times N$ images?
For filter design, it is useful to remember that the FOURIER transform of a Gaussian is a Gaussian.

FOURIER transform of a Gaussian is a Gaussian

First:

$$
\begin{aligned}
F(-2\pi jxf(x))(u) &= \int_{-\infty}^{+\infty} -2\pi jxf(x)e^{-2\pi jxu}dx = \int_{-\infty}^{+\infty} f(x)\frac{d}{du}(e^{-2\pi jxu})dx \\
&= \frac{d}{du}F(f(x))(u) = -2\pi jxF(f(x))(u)
\end{aligned}
$$

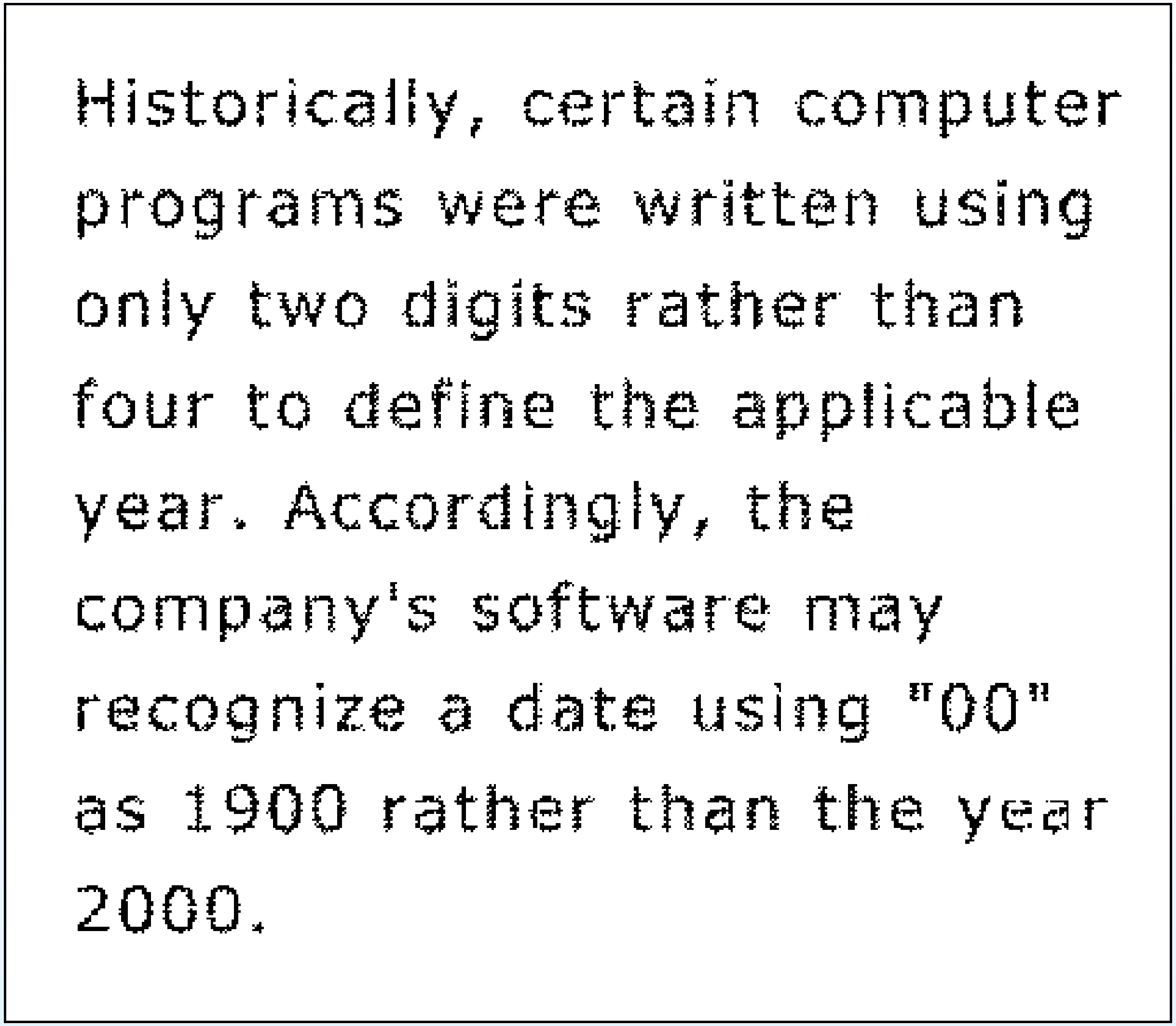
Then, if $f = e^{-ax^2}$ is a Gaussian (a is positive real): $\frac{df(x)}{dx} = -2axf(x)$ We take the FOURIER transform of both sides:

$$
\begin{aligned}
F(\frac{df(x)}{dx})(u) &= 2\pi juF(u) \text{ (derivative of a FOURIER transform)} \\
F(-2axf(x))(u) &= \frac{a}{\pi j}F(-2\pi jxf(x)) = \frac{a}{\pi j}\frac{d}{du}F(f(x))(u) \text{ (above)}
\end{aligned}
$$

putting things together:

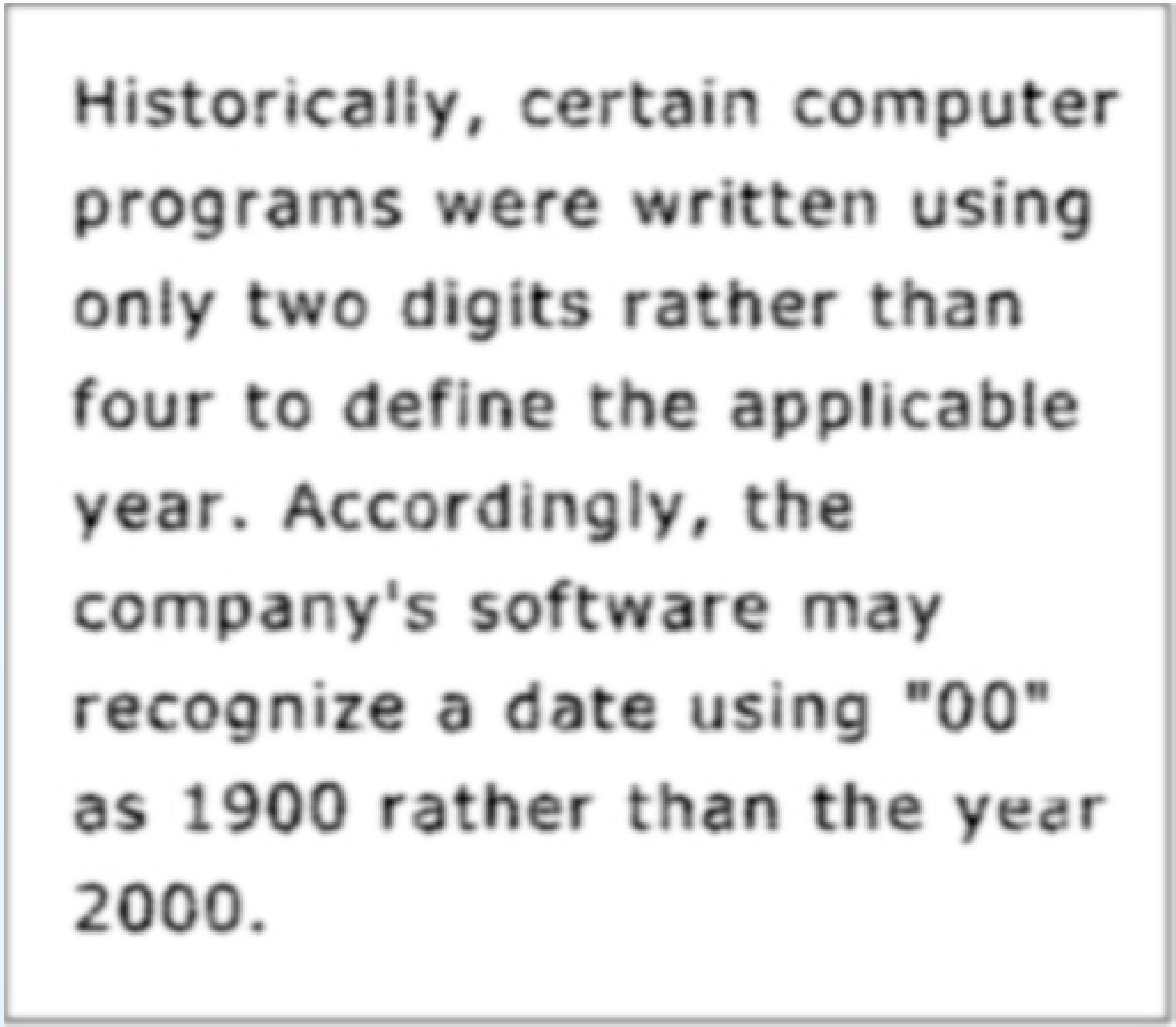
$$
\begin{aligned}
2\pi juF(f(x))(u) &= \frac{a}{\pi j}\frac{d}{du}(F(f(x))(u)) \\
\frac{d}{du}(F(f(x))(u)) &= -\frac{2\pi^2}{a}uF(f(x))(u) \Longrightarrow \Bigg\| F(u) = \sqrt{\frac{\pi}{a}}e^{-\frac{2\pi^2}{a}u^2}
\end{aligned}
$$

Example of simple convolution


Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.

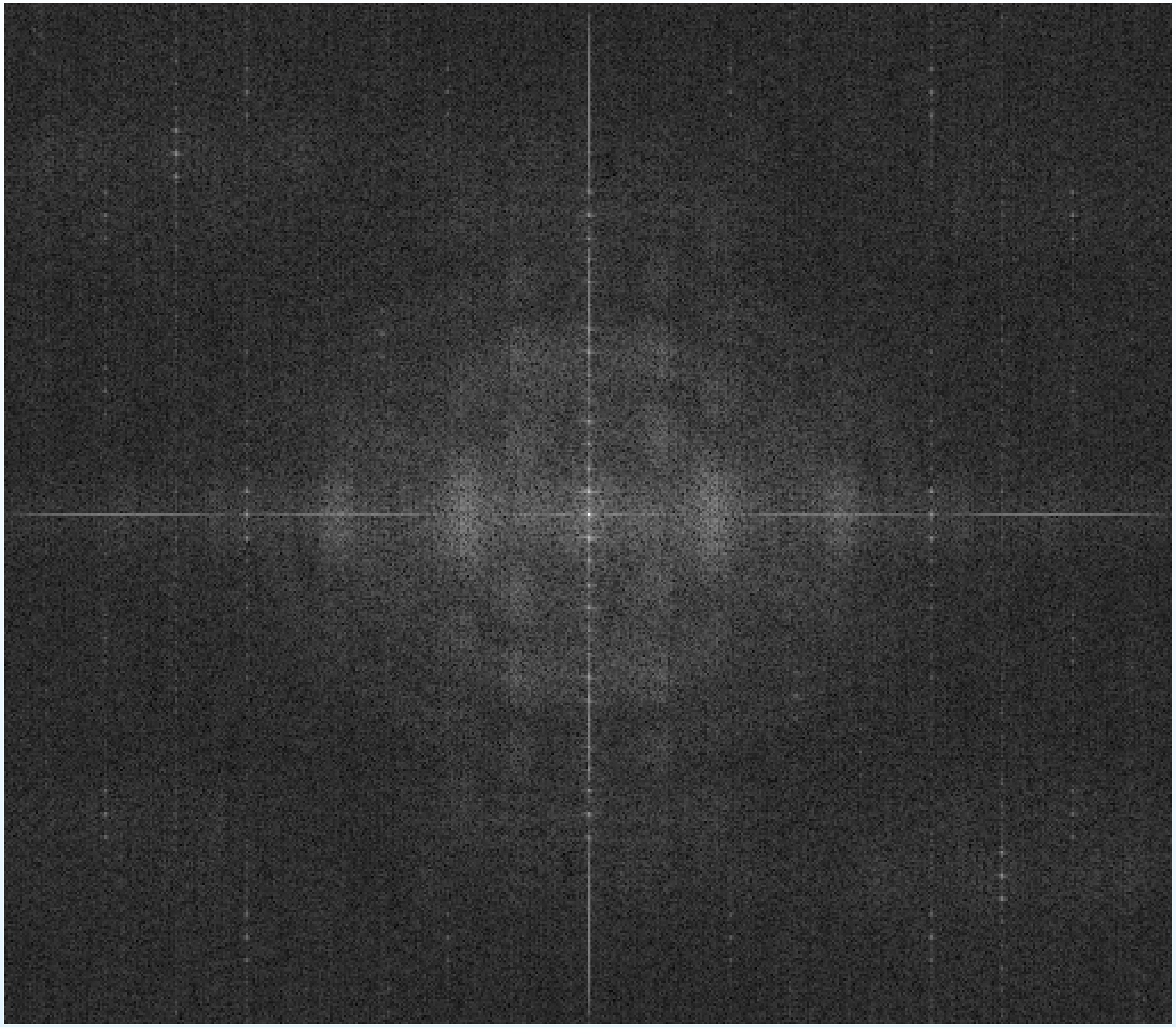
Corrupted text

Example of simple convolution



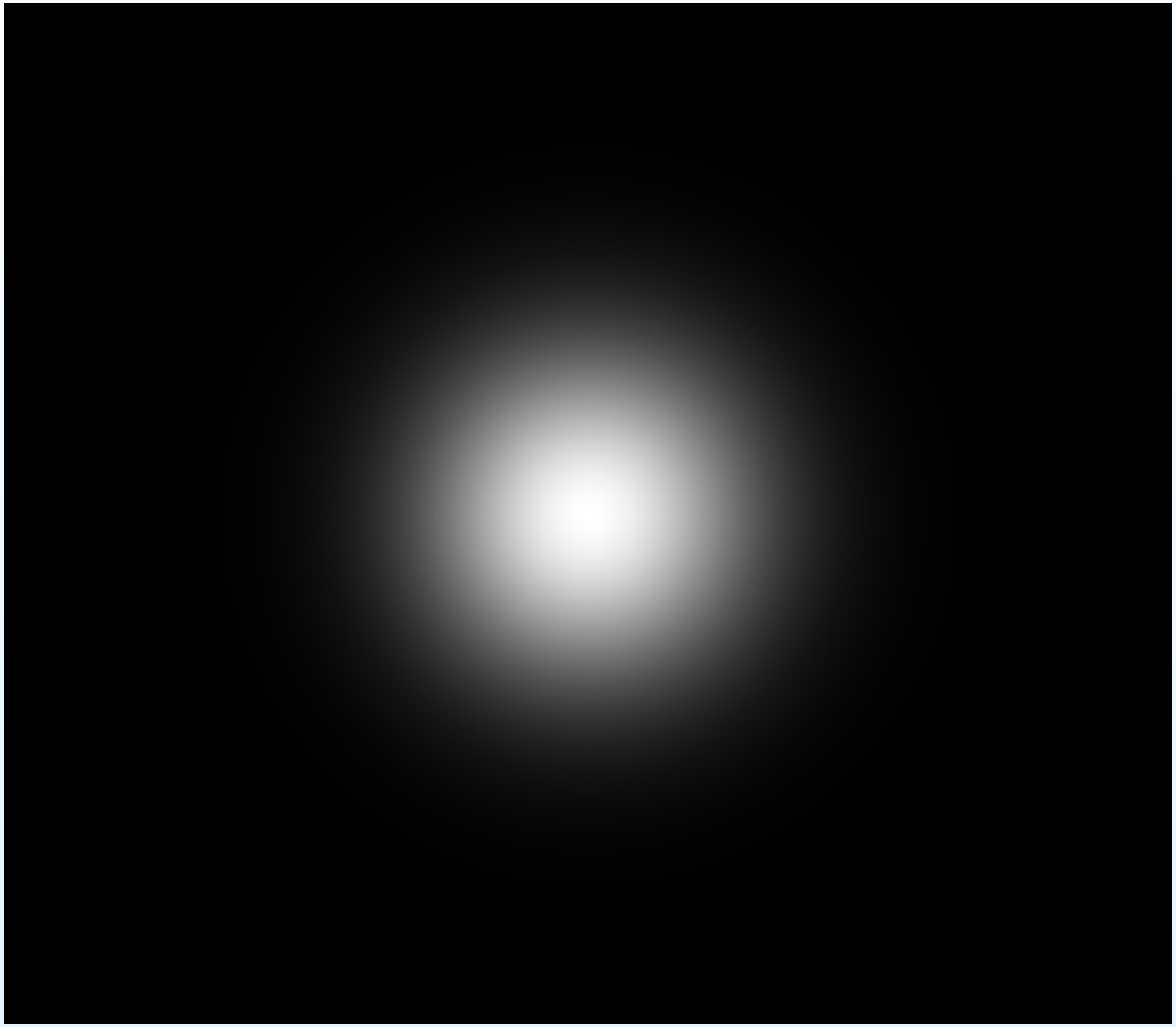
Filtered text

Example of simple convolution



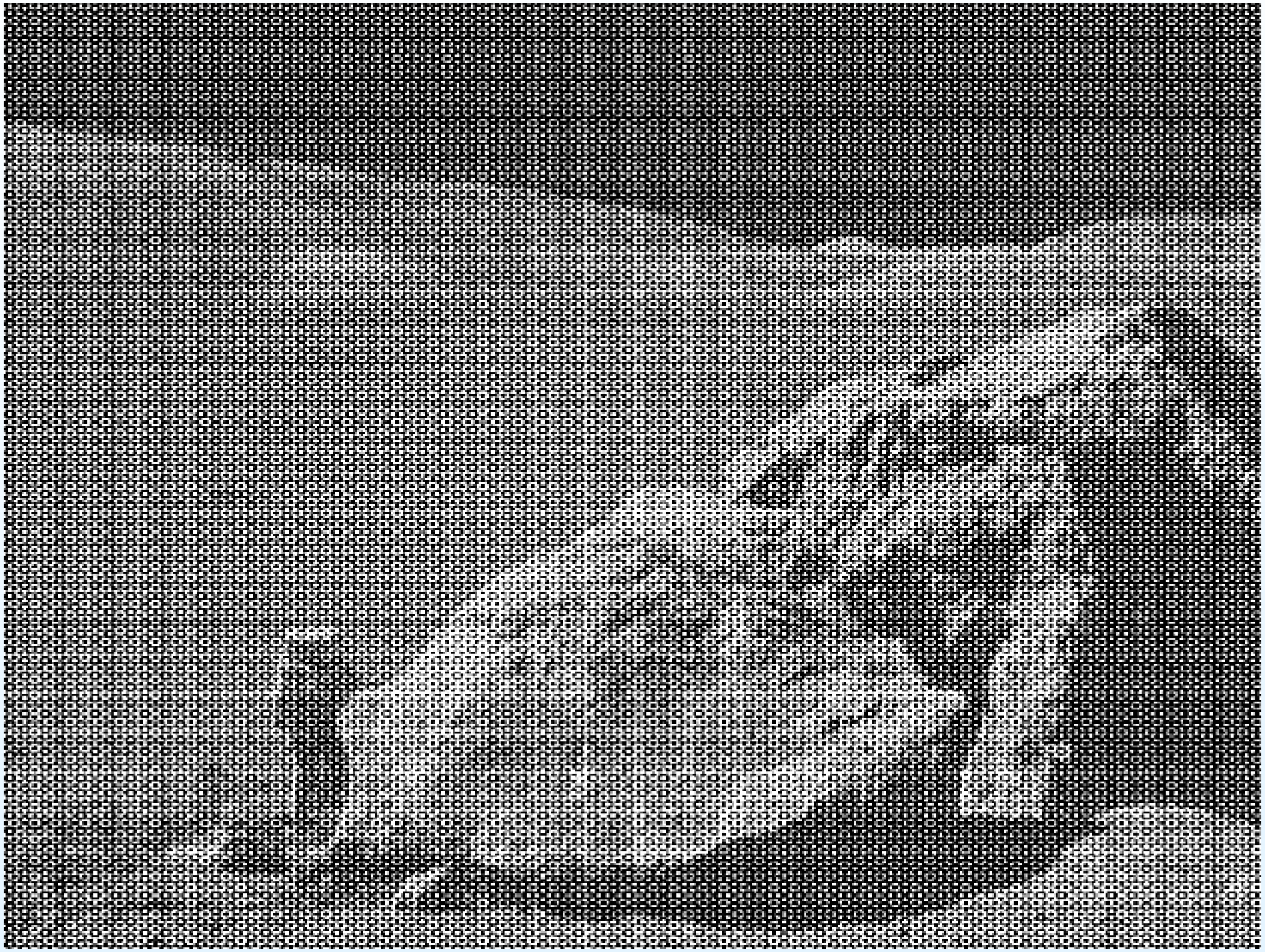
DFT of text

Example of simple convolution



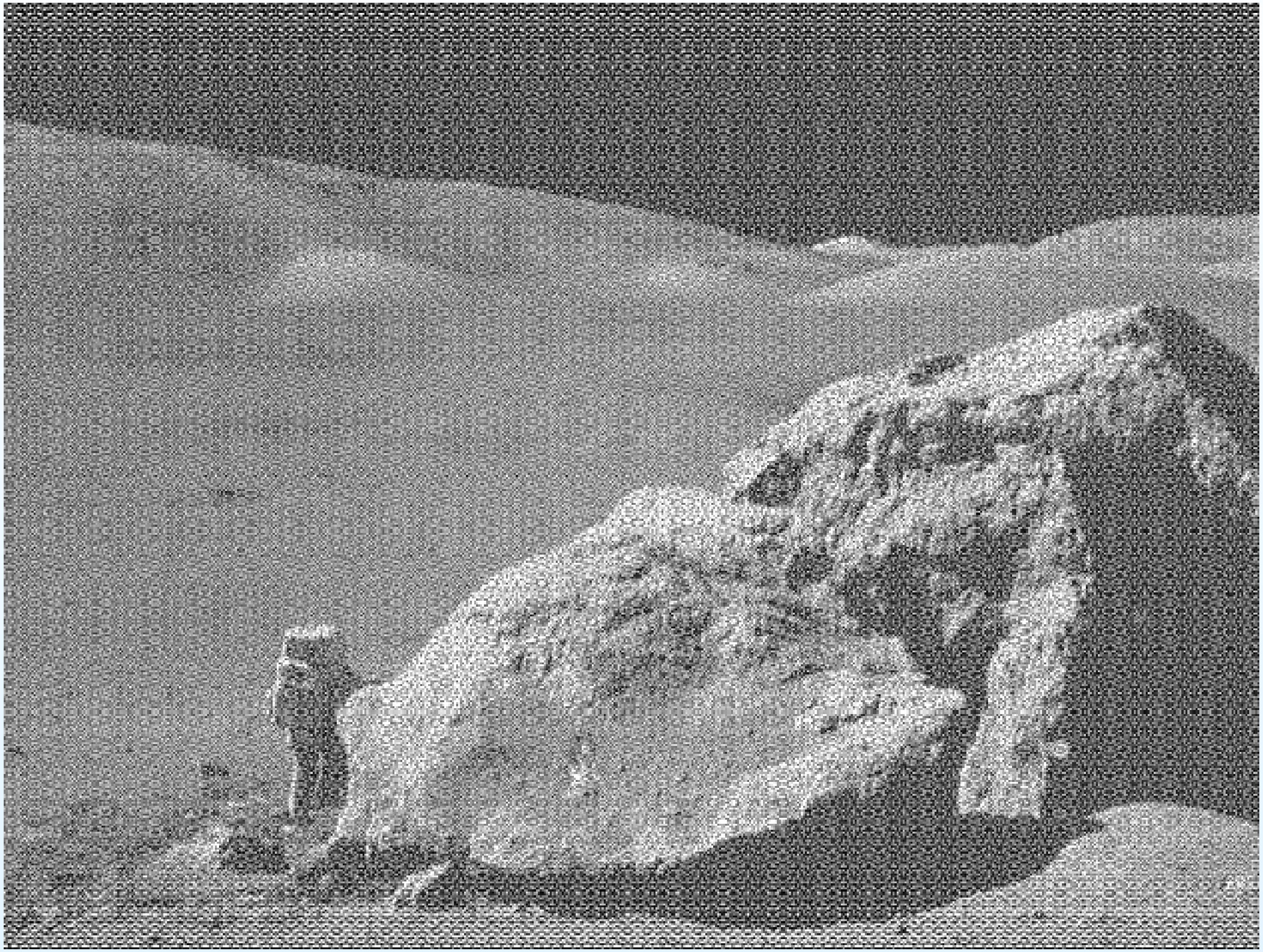
Low-pass filter using Gaussian

Example of non-trivial convolution



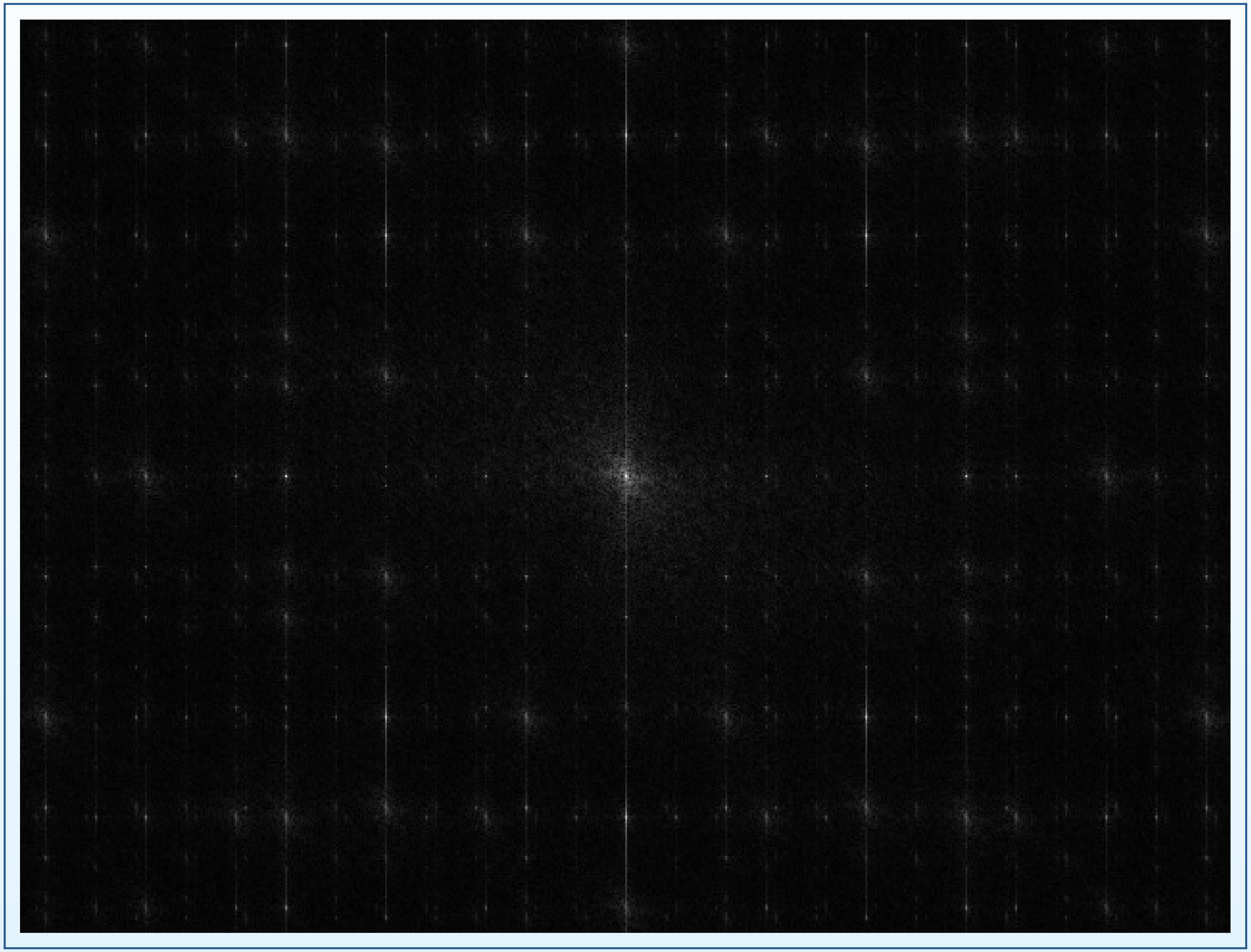
Corrupted Moon scene

Example of non-trivial convolution



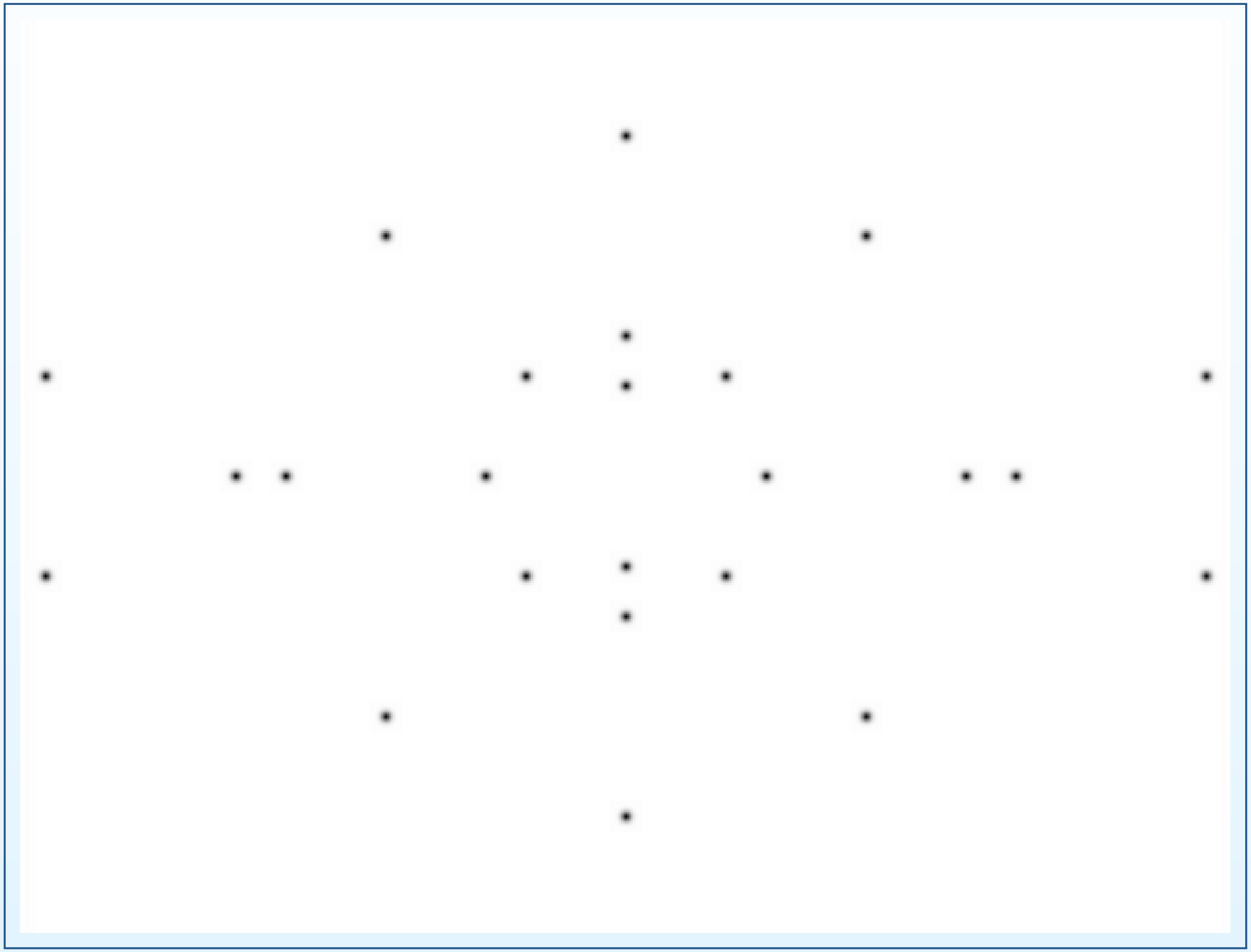
Filtered Moon scene

Example of non-trivial convolution



DFT of corrupt scene

Example of non-trivial convolution



Designed filter

The Fast Fourier Transform

History of the FFT

- Invented by C.F. Gauss in 1803, for doing astronomy-related hand calculation. Never published (found in notes).
- First modern algorithm attributed to Cooley and Tukey, 1963, IBM. IBM thought the algorithm was so important they decided to put it immediately in the public domain.
- Many implementations exist.
- A good one: FFTW, the Fastest FOURIER Transform in the West.

Decimation in time

Assume we want to do a DFT of a length which is a power of 2, i.e: $M = 2^n$. The DFT is written:

$$F(u) = \sum_{x=0}^{M-1} f(x) e^{-j(2\pi x u)/M}$$

We do the sums in two halves:

$$\begin{aligned} F(u) &= \sum_{x=0}^{\frac{M}{2}-1} f(2x) e^{-2j\pi(2x)u/M} + \sum_{x=0}^{\frac{M}{2}-1} f(2x+1) e^{-2\pi j(2x+1)u/M} \\ &= \sum_{x=0}^{\frac{M}{2}-1} f_{\text{even}}(x) e^{-2j\pi x u/(M/2)} + e^{-2\pi j u/M} . \sum_{x=0}^{\frac{M}{2}-1} f_{\text{odd}}(x) e^{-2j\pi x u/(M/2)} \end{aligned}$$

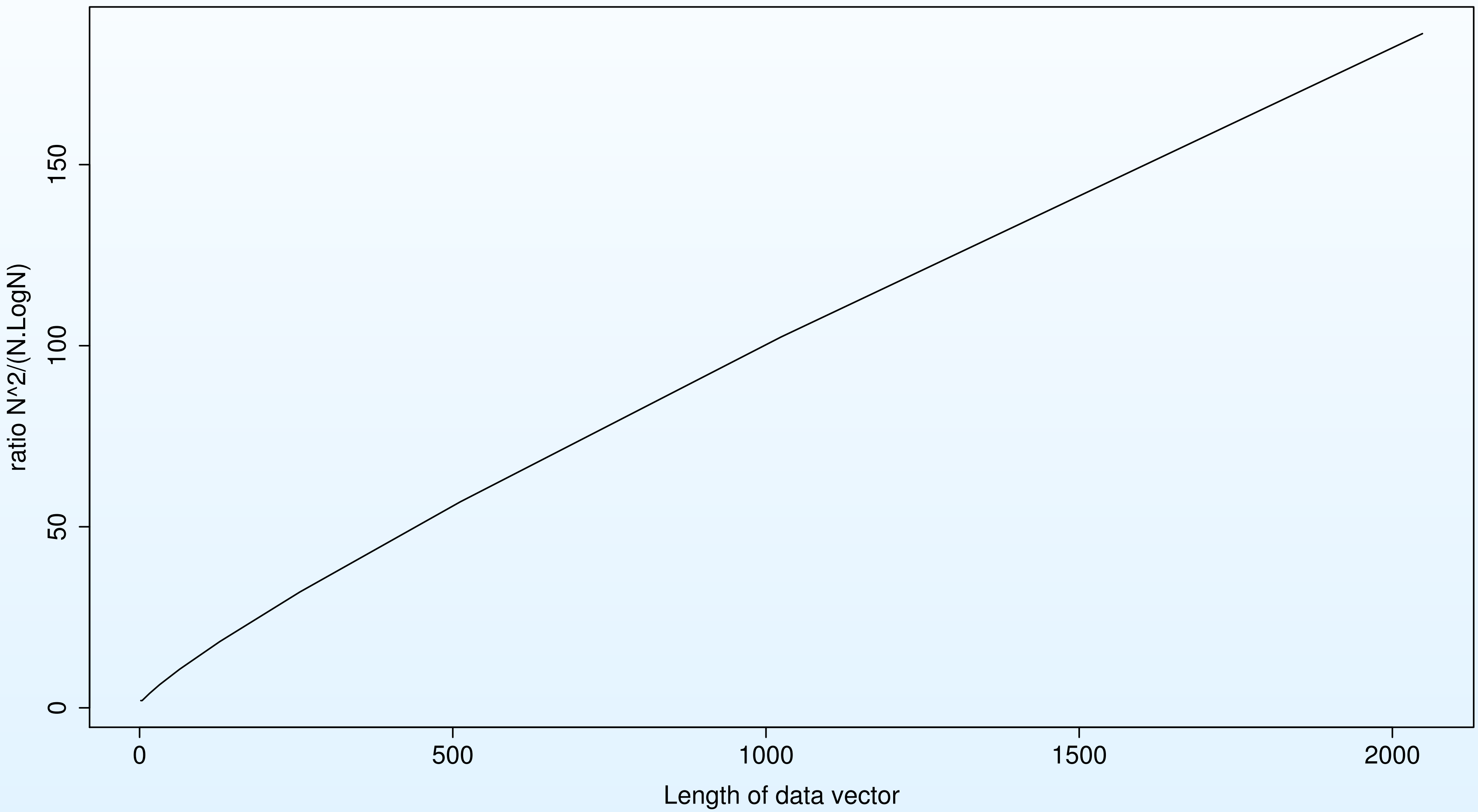
What have we gained?

- Seemingly nothing much, however...
- Now we are doing two $M/2$-length DFTs instead of a single M-length DFT.
- This means instead of doing M^2 operations, we do $2 \times (\frac{M}{2})^2 = \frac{M^2}{2}$ operations.
- Why stop here? Indeed we can decimate the two $M/2$ DFTs again, and so on recursively.
- In the final recursive structure, there are $n = \log_2(M)$ levels.
- At each levels there are M operations to perform.
- The final number of operations is

$$M. \log_2(M)$$

- Note: there are algorithms that are not limited to 2^n vector lengths (Singleton 1969).

What difference does it make?



Conclusion

What have we learned?

- Transforms are a change of basis representation. They allow to represent the *same* data in a different way.
- One very important tranform is the FOURIER transform and its discrete equivalent the DFT.
- The FOURIER transform allows users to represent the data in the frequency domain, like a prism for light.
- We can now do 1-D and 2-D DFTs
- The DFT allows users to compute convolution more quickly and easily.
- We learned how to do useful filterings using the DFTs do do enhancements using the frequency domain.
- There exists an efficient implementation of the DFT: the Fast FOURIER Transform, which makes all the previous operations all the more worthwhile.