A gentle introduction to Fourier analysis

Many slides borrowed from S. Seitz, A. Efros, D. Hoiem, B. Freeman, A. Zisserman
Mystery 1

- Why can downsampling sometimes lead to aliasing?
Mystery 2

- Why does filtering with a Gaussian give a nice smooth image, but filtering with a box filter gives artifacts?
Mystery 3

• How do hybrid images work?

“Low frequencies”

“High frequencies”

Salvador Dali
“Gala Contemplating the Mediterranean Sea, which at 30 meters becomes the portrait of Abraham Lincoln”, 1976
Fourier analysis

- To understand such phenomena, we need a representation of images that allows us to tease apart slow and fast changes
Outline

• Fourier series
• 1D Fourier transform
  • Definition and properties
  • Discrete Fourier transform
• 2D Fourier transform
  • Definition
  • Examples and properties
• Convolution theorem
• Understanding the sampling theorem
Fourier series

- Any periodic function on $[0, 1]$ can be expressed as a weighted sum of sinusoids of different frequencies (1807)

\[ \sum_{k=1,3,5,...}^{\infty} \frac{1}{k} \sin(kt) \]

Periodic means $f(0) = f(1)$

**= bunch of important details here

Example: series for a square wave

Jean-Baptiste Joseph Fourier (1768-1830)
Fourier series

Generally, we have for a (reasonable) periodic $f(t)$

$$f(t) \sim A_0 + \sum_{i=1}^{\infty} [A_i \cos (i2\pi t) + B_i \sin (i2\pi t)]$$

And we need to figure out the weights for a given $f(t)$. 
Fourier series: useful facts

\[ \int_0^1 \cos (i2\pi t) \, dt = \int_0^1 \sin (i2\pi t) \, dt = 0 \text{ for } i \text{ integer, } i > 0 \]  
Fact 1

\[ \int_0^1 \cos (i2\pi t) \sin (j2\pi t) \, dt = 0 \text{ for } i, j \text{ integer, } i \neq j, i > 0, j > 0 \]  
Fact 2

\[ \int_0^1 \cos (i2\pi t) \cos (j2\pi t) \, dt = 0 \text{ for } i, j \text{ integer, } i \neq j, i > 0, j > 0 \]  
Fact 2

\[ \int_0^1 \sin (i2\pi t) \sin (j2\pi t) \, dt = 0 \text{ for } i, j \text{ integer, } i \neq j, i > 0, j > 0 \]  
Fact 2

\[ \int_0^1 \sin^2 (i2\pi t) \, dt = 1/2 \text{ for } i \text{ integer} \]  
Fact 3

\[ \int_0^1 \cos^2 (i2\pi t) \, dt = 1/2 \text{ for } i \text{ integer} \]  
Fact 3
Fourier series: using facts

If:

\[ f(t) \sim A_0 + \sum_{i=1}^{\infty} \left[ A_i \cos(i2\pi t) + B_i \sin(i2\pi t) \right] \]

\[ \int_0^1 f(t) dt = A_0 \]

(fact 1 makes all the cosine/sine terms go away!)

\[ \int_0^1 f(t) \sin(i2\pi t) dt = \frac{A_i}{2} \]

(fact 2 makes all the other terms go away!)

\[ \int_0^1 f(t) \cos(i2\pi t) dt = \frac{B_i}{2} \]

And fact 3 sets the scale)
Fourier series: issues

- A’s and B’s are inelegant -> complex exponentials

- Did NOT show that the series converges to the function
  - Read Korner’s wonderful book Fourier Analysis
  - We’re OK for anything we care about

- In principle, we can go forward
  - Function -> A’s, B’s

- Or backward
  - A’s, B’s -> Function

- Is this right? (mostly yes, but details…)
Complex exponentials

This $i$ is the square root of -1 !!!

$$e^{i2k\pi t} = \cos(2k\pi t) + i \sin(2k\pi t)$$

$$f(t) \sim \sum_{k=0}^{\infty} c_k e^{i2k\pi t}$$

Advantage:
if the function is complex, can represent cleanly
don’t need to remember which is A, which B
Complex exponentials: compact facts

\[
\int_0^1 e^{i2k\pi t} e^{-i2n\pi t} dt = \begin{cases} 
0 & k \neq n \\
1 & k = n
\end{cases}
\]

k, n integers

This minus sign matters!
Fourier series with complex exponentials: using fact

If:
\[ f(t) \sim \sum_{k=0}^{\infty} c_k e^{i2k\pi t} \]

Using the fact! (this is analogous to an orthonormal basis in linear algebra)

\[ c_k = \int_{0}^{1} f(t) e^{-i2k\pi t} \, dt \]
Fourier series with complex exponentials: issues

• But this is just for a periodic function on $[0, 1]$
  • Easy to extend to other intervals
  • Easy to extend to the circle

• But what about functions on $[-\infty, \infty]$?
  • These could wiggle often in numerous places
  • IDEA: use “more” basis elements

• The Fourier transform
1D Fourier transform

- Let’s define an (overcomplete) set of basis functions:
  \[ \psi_u(t) = e^{i2\pi ut}, \quad u \in (-\infty, \infty) \]

- Compare

\[ e^{i2k\pi t} = \cos(2k\pi t) + i \sin(2k\pi t) \]
1D Fourier transform

• Let’s define a (continuously parameterized) set of basis functions:

\[ \psi_u(t) = e^{i 2\pi u t}, \quad u \in (-\infty, \infty) \]

• Inner product for complex functions is given by:

\[ \langle f, g \rangle = \int_{-\infty}^{\infty} f(t) g^*(t) dt \]

Complex conjugate: real part stays the same, imaginary part is flipped
1D Fourier transform

- Let’s define a (continuously parameterized) set of basis functions:
  \[ \psi_u(t) = e^{i2\pi ut}, \quad u \in (-\infty, \infty) \]

- Inner product for complex functions is given by:
  \[ \langle f, g \rangle = \int_{-\infty}^{\infty} f(t)g^*(t)dt \]

- Orthonormality:
  \[ \langle \psi_{u_1}, \psi_{u_2} \rangle = \begin{cases} 1 & \text{if } u_1 = u_2 \\ 0 & \text{otherwise} \end{cases} \]
1D Fourier transform

- Given a signal $f(t)$, we want to represent it as a weighted combination of the basis functions $\psi_u(t) = e^{i2\pi ut}$ with weights $F(u)$:

$$f(t) = \int_{-\infty}^{\infty} F(u)e^{i2\pi ut} du$$

- Each weight $F(u)$ is given by the inner product of $f$ and $\psi_u$:

$$F(u) = \langle f, \psi_u \rangle = \int_{-\infty}^{\infty} f(t)e^{-i2\pi ut} dt$$
1D Fourier transform

- Forward transform: \[ f(t) \xrightarrow{\mathcal{F}} F(u) \]

\[ F(u) = \int_{-\infty}^{\infty} f(t) e^{-i2\pi ut} dt \]

- Note: for the FT to exist, the energy \( \int_{-\infty}^{\infty} |f(t)|^2 \, dt \) has to be finite
1D Fourier transform

- Forward transform: $f(t) \xrightarrow{\mathcal{F}} F(u)$

$$F(u) = \int_{-\infty}^{\infty} f(t)e^{-i2\pi ut} dt$$

- For each $u$, $F(u)$ is a complex number that encodes both the amplitude $A$ and phase $\phi$ of the sinusoid $A \sin(2\pi ut + \phi)$ in the decomposition of $f(t)$:

$$F(u) = \text{Re}(F(u)) + i \text{Im}(F(u)),$$

$$A = \sqrt{\text{Re}(F(u))^2 + \text{Im}(F(u))^2}, \quad \phi = \tan^{-1}\frac{\text{Im}(F(u))}{\text{Re}(F(u))}$$

- If $f(t)$ is real, then $\text{Re}(F(u)) = \text{Re}(F(-u))$, $\text{Im}(F(u)) = -\text{Im}(F(-u))$
1D Fourier transform

- Forward transform: $f(t) \xrightarrow{\mathcal{F}} F(u)$

  $$F(u) = \int_{-\infty}^{\infty} f(t) e^{-i2\pi ut} \, dt$$

- Important properties:
  - Energy preservation:
    $$\int_{-\infty}^{\infty} |f(t)|^2 \, dt = \int_{-\infty}^{\infty} |F(u)|^2 \, du$$
  - Linearity: $\mathcal{F}\{af_1 + bf_2\} = a\mathcal{F}\{f_1\} + b\mathcal{F}\{f_2\}$

Parseval's Theorem!
1D Fourier transform

- **Forward transform:** \[ f(t) \xrightarrow{\mathcal{F}} F(u) \]
  \[
  F(u) = \int_{-\infty}^{\infty} f(t) e^{-i2\pi ut} dt
  \]

- **Inverse transform:** \[ F(u) \xrightarrow{\mathcal{F}^{-1}} f(t) \]
  \[
  f(t) = \int_{-\infty}^{\infty} F(u) e^{i2\pi ut} du
  \]

- **Duality:** if \( f(t) \xrightarrow{\mathcal{F}} F(u) \), then \( F(t) \xrightarrow{\mathcal{F}} f(-u) \)
  - Thus, we can talk about *Fourier transform pairs* \( f(t) \leftrightarrow F(u) \)
Important Fourier transform pairs

\[ \text{box}(t) \]

\[ \text{sinc}(u) = \frac{\sin(\pi u)}{\pi u} \]
Important Fourier transform pairs

\[ \text{box}(t) \]

\[ \text{sinc}(u) = \frac{\sin(\pi u)}{\pi u} \]

\[ \text{gauss}(t; \sigma) \]

\[ \text{gauss}(u; \frac{1}{\sigma}) \]
Important Fourier transform pairs

\[
\text{box}(t) = \begin{cases} 
1 & \text{for } |t| \leq 0.5 \\
0 & \text{otherwise}
\end{cases}
\]

\[
sinc(u) = \frac{\sin(\pi u)}{\pi u}
\]

\[
\text{gauss}(t; \sigma) = e^{-\frac{(t-\mu)^2}{2\sigma^2}}
\]

\[
\text{gauss}(u; \frac{1}{\sigma}) = e^{-\frac{u^2}{2\sigma^2}}
\]

\[
f(t) = 1
\]

\[
\text{unit impulse } \delta(u)
\]

*The last one is formal since these functions don’t meet the mathematical requirements for FT*
Important Fourier transform pairs

\[ \text{box}(t) \]

\[ \text{sinc}(u) = \frac{\sin(\pi u)}{\pi u} \]

\[ \text{gauss}(t; \sigma) \]

\[ \text{gauss}(u; \frac{1}{\sigma}) \]

\[ f(t) = 1 \]

\[ \text{unit impulse } \delta(u) \]

Notice that when f has narrower support, FT(f) has broader, and Vice versa!

*The last one is formal since these functions don’t meet the mathematical requirements for FT*
Outline

• 1D Fourier transform
  • Definition and properties
  • Discrete Fourier transform
Discrete Fourier transform (DFT)

- Now suppose our signal consists of $N$ samples $f(n)$, $n = 0, ..., N - 1$
- We can also discretize frequencies to $k/N$, $k = 0, ..., N - 1$ ($k$ cycles per $N$ samples)
Discrete Fourier transform (DFT)

• Now suppose our signal consists of $N$ samples $f(n)$, $n = 0, ..., N - 1$

• We can also discretize frequencies to $k/N$, $k = 0, ..., N - 1$ ($k$ cycles per $N$ samples)

• DFT formula:

$$F(k) = \sum_{n=0}^{N-1} f(n)\exp\left(-i\frac{2\pi k}{N} n\right)$$

• We can pack the values $\exp\left(-i\frac{2\pi k}{N} n\right)$, $k, n = 0, ..., N - 1$ into an $N \times N$ matrix $U$, and DFT becomes just a matrix-vector multiplication!

• **Fast Fourier transform**: only $N \log N$ complexity!
DFT: Just a change of basis!

\[ U f = F \]
Inverse DFT

- **Forward DFT:**
  \[
  F(k) = \sum_{n=0}^{N-1} f(n) \exp\left(-i \frac{2\pi}{N} kn\right) \quad \text{or} \quad F = Uf
  \]

- **Inverse DFT:**
  \[
  f(n) = \frac{1}{N} \sum_{n=0}^{N-1} F(k) \exp\left(i \frac{2\pi}{N} kn\right) \quad \text{or} \quad f = \frac{1}{N} U^{-1}F
  \]

where \( U^{-1} \) is the transpose of the complex conjugate of \( U \)
Periodicity of DFT and inverse DFT

• The result of DFT is periodic: because $F(k)$ is obtained as a sum of complex exponentials with a common period of $N$ samples, $F(k + aN) = F(k)$ for any integer $a$:

$$F(k + aN) = \sum_{n=0}^{N-1} f(n) \exp \left( -i \frac{2\pi}{N} n(k + aN) \right)$$

$$= \sum_{n=0}^{N-1} f(n) \exp \left( -i \frac{2\pi n}{N} k \right) \exp(-i2\pi an) = F(k)$$

• Likewise, the result of the inverse DFT is a periodic signal: $f(t + aN) = f(t)$ for any integer $a$
Outline

• 1D Fourier transform
  • Definition and properties
  • Discrete Fourier transform

• 2D Fourier transform
2D Fourier transform

- To represent 2D signals $f(x, y)$, we need to extend our 1D basis functions $\psi_u(t) = e^{i2\pi ut}$ to two variables:

$$\psi_{u,v}(x, y) = e^{i2\pi ux}e^{i2\pi vy} = e^{i2\pi (ux+vy)}$$
$$= \cos 2\pi (ux + vy) + i \sin 2\pi (ux + vy)$$

- What does this look like?
2D Fourier transform

- 2D basis functions are oriented sinusoidal “gratings”:
  - $(u, v)$ is the direction normal to the grating
  - The period is $1/\sqrt{u^2 + v^2}$
Basis function examples

\((u, v)\)

Real component
Basis function examples

\((u, v)\)

Real component
Basis function examples

\[(u, v)\]

Real component
Linear combination of basis functions

\[(u, v)\]

Real component
2D Fourier transform

\[ F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-i2\pi(ux + vy)} dx \, dy \]

- Output is 2D and complex-valued:
  \[ F(u, v) = \text{Re}(F(u, v)) + i \text{Im}(F(u, v)) \]
- Magnitude spectrum: \[ |F(u, v)| = \sqrt{\text{Re}(F(u, v))^2 + \text{Im}(F(u, v))^2} \]
- Phase angle spectrum: \[ \tan^{-1} \left( \frac{\text{Im}(F(u,v))}{\text{Re}(F(u,v))} \right) \]
- Symmetry: the Fourier transform of a real-valued image has coefficients that come in pairs, with \( F(u, v) \) being the complex conjugate of \( F(-u, -v) \)
  - This means that the magnitude spectrum is symmetric about the origin
2D discrete Fourier transform

\[ F(u, v) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} f(n, m) \exp \left( -i2\pi \left( \frac{un}{N} + \frac{vm}{M} \right) \right) \]

Source: B. Freeman
Real image examples
Real image examples

intensity image

log fft magnitude
Which image goes with which spectrum?
Phase vs. magnitude

- Which has more information, the phase or the magnitude?
- Let’s take the phase from one image and combine it with the magnitude from another image.
Images with periodic patterns

- The magnitude image has peaks corresponding to the frequencies of repetition

Source: A. Zisserman
Application: Removing periodic patterns

Source: A. Zisserman
Periodic patterns

Lunar orbital image (1966)  Magnitude image

Why are there multiple peaks in the magnitude image?
Application: Removing periodic patterns

Lunar orbital image (1966)  Magnitude image  Remove peaks  Join lines removed

You should think of this as a kind of local smoothing, but in the Fourier domain!

Source: A. Zisserman
**Image transformations**

- How does the FT change when the image is scaled?

Scaled by the inverse factor!
In 1D

2D is easy, follows this form

\[ \mathcal{F}(f(at)) = \int_{-\infty}^{\infty} f(at) \exp[-i2\pi ut] \, dt \]

\[ = \frac{1}{a} \int_{-\infty}^{\infty} f(s) \exp[-i2\pi u/as] \, dt \]

\[ = \frac{1}{a} \mathcal{F}(f)(u/a). \]
Important effect

“wider” function has “narrower” Fourier transform

“narrower” function has “wider” Fourier transform

FIGURE 7.1: Top shows $f(t)$ and its magnitude spectrum, and bottom $f(2t)$ and its magnitude spectrum. Notice how narrowing the function broadens the Fourier transform (from top to bottom); or broadening it narrows the Fourier transform (from bottom to top).
Image transformations

• How does the FT change when the image is rotated?

Rotates the same way!
In 2D

(both fairly easy to prove)

<table>
<thead>
<tr>
<th>for $\mathcal{R}$ a rotation matrix $f(\mathcal{R}x)$</th>
<th>$\mathcal{F}(f)(\mathcal{R}u)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>for $\mathcal{A}$ an invertible matrix and $z$ a constant vector $f(\mathcal{A}x + w)$</td>
<td>$\frac{1}{\det(\mathcal{A})} \exp[-i2\pi u^T w] \mathcal{F}(f)(\mathcal{A}^{-T} u)$</td>
</tr>
</tbody>
</table>
Image transformations

- How does the FT change when the image is rotated?

Caution: in real images this is not always the case because of edge artifacts (recall that DFT treats images as periodically tiled)
Image transformations

• How does the FT change when the image is translated?

Magnitude spectrum doesn’t change, phase gets modulated.
Translation in 1D

ND works the same way

To see this, we need one more result. Write $\text{shift}_\tau$ for the operation that maps the function $f(t)$ to the function $\text{shift}_\tau(f) = f(t - \tau)$. Then

$$\mathcal{F}(\text{shift}_\tau f) = \mathcal{F}(f) \exp[-i2\pi u \tau]$$

because

$$\int_{-\infty}^{\infty} f(t - \tau) \exp[-i2\pi ut] \, dt = \int_{-\infty}^{\infty} f(w) \exp[-i2\pi u(w + \tau)] \, dw$$

$$= \int_{-\infty}^{\infty} f(w) \exp[-i2\pi uw] \, dw \exp[-i2\pi u\tau]$$

$$= [\mathcal{F}(f)] \exp[-i2\pi u\tau]$$
A model of Sampling

We want to model sampling in a way that allows us to take Fourier Transforms.

Challenges:

Should be able to compute a meaningful integral of the sampled data

In particular, we would like

$$\int_W \text{sample}_1D(f(t))g(t)dt$$

(where $W$ is some interval) to be as similar as possible to

$$\int_W f(t)g(t)dt.$$
A trick – the delta function

Define the delta function in 1D by

\[
\delta(x) = \begin{cases} 
0 & x \neq 0 \\
\text{uncomfortable} & x = 0
\end{cases}
\]

\[
\int f(x)\delta(x)\,dx = f(0)
\]

This isn’t a function in any familiar sense, but it is useful and crops up in all sorts of places.
Sampling in 1D

Remember this: The proper mathematical model to use for sampling in 1D is:

$$\text{sample}_{1D}(f(t)) = \sum_{i=-\infty}^{\infty} f(i)\delta(t - i).$$

It is a straightforward exercise to modify this model so that samples occur at discrete points that aren’t integer points (just fiddle with the indexing).
Sampling in 1D

**FIGURE 7.2:** Sampling in 1D takes a function $f(t)$ and returns a set of weighted delta functions at each sample point, $\sum_{i=-\infty}^{\infty} f(i) \delta(t - i)$. It is usual to draw delta functions as arrows, with the height of the arrow determined by the weight of the delta function, even though the value of the delta function is hard to talk about.
Delta functions in 2D

Sampling in 2D is very like sampling in 1D. We need a delta function in two dimensions (which behaves a lot like one in 1D). We have \( \delta(x - s, y - t) \) is zero at all points where \( x \neq s \) or \( y \neq t \). We do not discuss the value at the crucial point; instead, the function has the property

\[
\int_{-\infty}^{\infty} f(x, y) \delta(x - s, y - t) \, dx \, dy = f(s, t).
\]
Sampling in 2D

We now use the model

$$\text{sample}_{2D}(f(t)) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} f(i, j) \delta(x - i, y - j).$$

The properties of the delta function can be used to establish that

$$\int_{W} \text{sample}_{2D}(f(x, y)) g(x, y) dt = \sum_{i \in W} f(i) g(i)$$

(where $i \in W$ refers to integer points in the interval). Notice that whether this is an accurate estimate of the integral or not depends a lot on how $f$ and $g$ behave – if they don’t vary much between sample points, the estimate will be good, and if there is a lot of variation between sample points, then the estimate will be poor. This should be true of a sampled function.
Sampling in 2D

FIGURE 7.4: Sampling in 2D takes a function and returns an array; again, we allow the array to be infinite dimensional and to have negative as well as positive indices.
Outline

• 1D Fourier transform
  • Definition and properties
  • Discrete Fourier transform

• 2D Fourier transform
  • Definition
  • Examples and properties

• Convolution theorem
Convolution theorem

- **Convolution** in the spatial domain translates to **multiplication** in the frequency domain (and vice versa).

- The Fourier transform of the convolution of two functions is the product of their Fourier transforms:
  \[ \mathcal{F}\{f * g\} = \mathcal{F}\{f\} \mathcal{F}\{g\} \]

- The inverse Fourier transform of the product of two Fourier transforms is the convolution of the two inverse Fourier transforms:
  \[ \mathcal{F}^{-1}\{FG\} = \mathcal{F}^{-1}\{F\} * \mathcal{F}^{-1}\{G\} \]
2D convolution theorem example

Image

Filter

Filtered image

FT(Image)

FT(Filter)

FT(Filtered image)
Easy to prove

\[
\mathcal{F}(f \ast g) = \int_{-\infty}^{\infty} \left( \int_{-\infty}^{\infty} f(t - \tau)g(\tau) \, d\tau \right) \exp \left[ -i2\pi ut \right] \, dt
\]

\[
= \int_{-\infty}^{\infty} \left( \int_{-\infty}^{\infty} f(t - \tau) \exp \left[ -i2\pi ut \right] \, dt \, dg(\tau) \right) \, d\tau
\]

\[
= \int_{-\infty}^{\infty} \left[ \mathcal{F}(f) \right] \exp \left[ -i2\pi ut \right] g(\tau) \, d\tau
\]

\[
= \left[ \mathcal{F}(f) \right] \int_{-\infty}^{\infty} g(\tau) \exp \left[ -i2\pi u\tau \right] \, d\tau
\]

\[
= \left[ \mathcal{F}(f) \right] \left[ \mathcal{F}(g) \right]
\]
Convolution theorem

• Suppose $f$ and $g$ both consist of $N$ pixels
• What is the complexity of computing $f * g$ in the spatial domain?
  • $O(N^2)$
• And what is the complexity of computing $\mathcal{F}^{-1}\{\mathcal{F}\{f\}\mathcal{F}\{g\}\}$?
  • $O(N \log N)$ using FFT
• Thus, convolution of an image with a large filter can be more efficiently done in the frequency domain
Understanding the behavior of filtering

• Why does filtering with a Gaussian give a nice smooth image, but filtering with a box filter gives artifacts?
Recall: Fourier transform pairs

\[ \text{box}(t) \cdot \text{sinc}(u) = \frac{\sin(\pi u)}{\pi u} \]

\[ \text{gauss}(t; \sigma) \cdot \text{gauss}(u; \frac{1}{\sigma}) \]
Filtering with a Gaussian
Filtering with a box filter
Low-pass and high-pass filtering

Image  | Low-pass filtered | High-pass filtered
---|---|---
![Image](source)

Source: A. Zisserman
Closer look at low-pass filtering

• Do we like this low-pass filtering result?
• No – it causes ringing artifacts in the image (why?)
  • Recall: it’s equivalent to convolving with a sinc function in the spatial domain
• This is why Gaussian filtering is preferred
Hybrid images in the frequency domain

1- Frequency cut = 16 cycles/image

2- Frequency cut = 36 cycles/image

Frequency domain

Source
Human contrast sensitivity curve

- Depending on viewing distance, peak sensitivity will occur at different frequencies
Outline

• 1D Fourier transform
  • Definition and properties
  • Discrete Fourier transform

• 2D Fourier transform
  • Definition
  • Examples and properties

• Convolution theorem

• Understanding the sampling theorem
Understanding sampling and aliasing
Recall: Nyquist-Shannon sampling theorem

- When sampling a signal at discrete intervals, the sampling frequency must be at least \textit{twice} the maximum frequency of the input signal to allow us to reconstruct the original perfectly from the sampled version.

Understanding the sampling theorem

- Suppose we have a continuous function \( f(t) \) and we want to sample it at discrete intervals with a spacing of \( T \).
- This can be accomplished by multiplying it by the *comb function* or *impulse train*:

\[
\text{comb}(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT)
\]

\[
f_s(t) = \sum_{n=-\infty}^{\infty} f(t) \delta(t - nT)
\]

Source: A. Zisserman
Understanding the sampling theorem

- Let’s (formally) take the Fourier transform:

\[ f_s(t) = f(t) \times \text{comb}(t; T) \]

\[ F_s(u) = F(u) \ast \frac{1}{T} \text{comb}\left(t; \frac{1}{T}\right) \]

*Officially, the FT of the comb function doesn’t exist since it’s periodic, and since \( \delta \) is a weird function

Source: A. Zisserman
Understanding the sampling theorem

- Let’s (formally) take the Fourier transform:

\[ f_s(t) = f(t) \times \text{comb}(t; T) \]

\[ F_s(u) = F(u) \ast \frac{1}{T} \text{comb} \left( t; \frac{1}{T} \right) \]

= Replicated copies of \( F(u) \)!

Source: A. Zisserman
Understanding the sampling theorem

• How do we reconstruct $f(t)$?
• Let’s apply a box filter in the frequency domain (equivalent to convolving with a sinc function in the original domain)
• When will this succeed?
  • When the sampling frequency $1/T$ exceeds twice the greatest frequency contained in $F(u)$!

Source: A. Zisserman
Understanding the sampling theorem

- If the sampling frequency is too small, frequencies above the Nyquist limit are “folded back” onto smaller frequencies, resulting in aliasing.

Source: A. Zisserman
Sampling theorem in 2D

- If the Fourier transform of a continuous function $f(x, y)$ is zero for all frequencies beyond $u_b$ and $v_b$, i.e., if the Fourier transform is band-limited, then $f(x, y)$ can be completely reconstructed from its samples as long as the sampling distances $w$ and $h$ along the $x$ and $y$ directions are such that $w \leq \frac{1}{2u_b}$ and $h \leq \frac{1}{2v_b}$

Source: A. Zisserman
Aside: Analyzing interpolation methods

- Perfect reconstruction of the subsampled signal requires convolution with a sinc filter in the spatial domain, which is bad because sinc has infinite support.
- Instead, simpler reconstruction (interpolation) methods are typically used.
Aside: Analyzing different interpolation methods

- Linear reconstruction can be done by convolving the sampled signal with a triangle filter:

- However, the Fourier transform of the triangle filter is the \( \text{sinc}^2 \) function, so multiplying the signal’s spectrum by it introduces high-frequency artifacts.
Bilinear interpolation closeup
Why else should you care about Fourier analysis?

Figure 7: Frequency analysis on each dataset. We show the average spectra of each high-pass filtered image, for both the real and fake images, similar to Zhang et al. [50]. We observe periodic patterns (dots or lines) in most of the synthetic images, while BigGAN and ProGAN contains relatively few such artifacts.

S.-Y. Wang et al. CNN-generated images are surprisingly easy to spot... for now. CVPR 2020
Why else should you care about Fourier analysis?

Checkerboard and repetition artifacts in GAN-generated images

https://distill.pub/2016/deconv-checkerboard/
Smoothing and downsampling

<table>
<thead>
<tr>
<th>256x256</th>
<th>128x128</th>
<th>64x64</th>
<th>32x32</th>
<th>16x16</th>
</tr>
</thead>
</table>

![Image showing different resolutions and smoothing effects](image-url)
Smoothing and downsampling

Smoothed by gaussian sigma 1 pixel
Smoothing and downsampling

Smoothing by gaussian sigma 2 pixels
What smoothing filter to use?

Why does smoothing with a Gaussian help?

Is there a better filter than a Gaussian?

Can you resample without loss?