Introduction to Fourier Analysis – Part 2

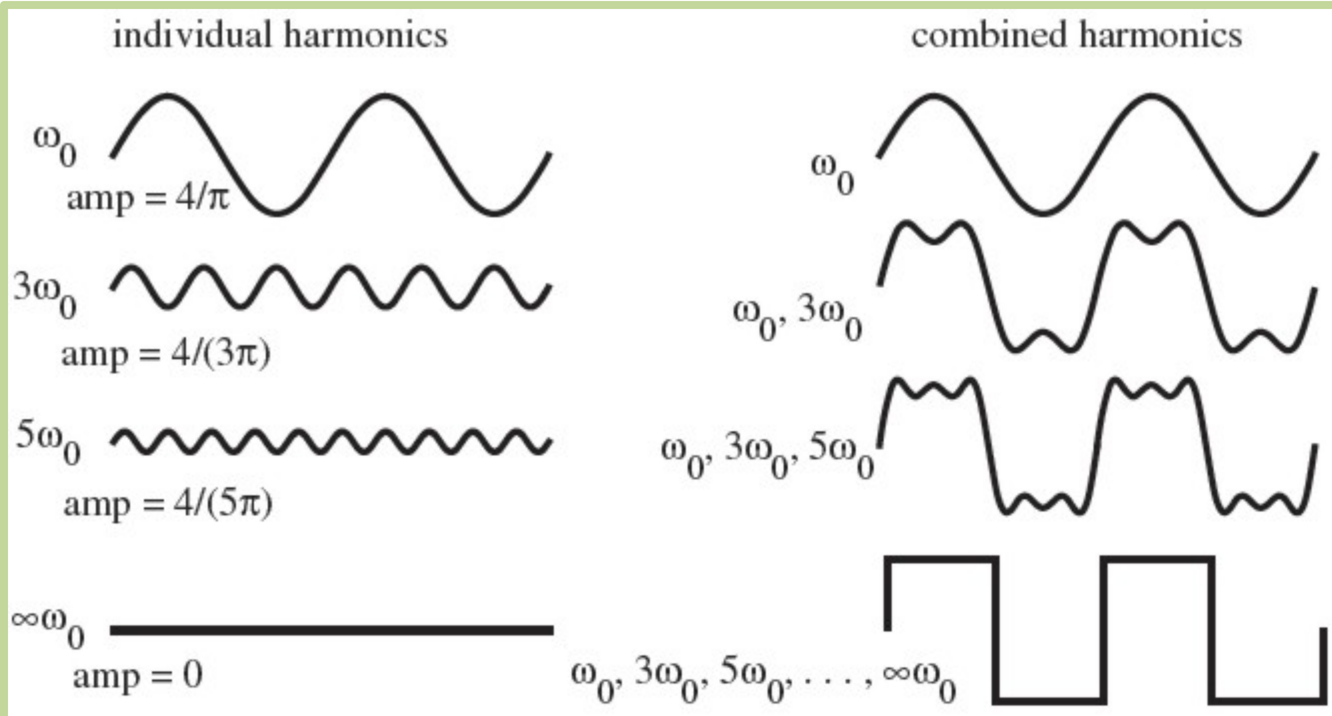
CS 510

Lecture #7

February 13, 2019

Colorado State University

In the extreme, a square wave



Graphic from *http://www.mechatronics.colostate.edu/figures/4-4.jpg*

Fourier Transform

- Formally, the Fourier transform in 1D is:

$$F(u) = \int_{-\infty}^{+\infty} f(x)\left[\cos 2\pi ux - i\sin 2\pi ux\right]dx$$

Where:

u is an integer in the range from 0 to ∞

–i is used to create a 2D vector space

$F(u) = a_u + ib_u$

Inverse Fourier Transform

- What if I have *F*(u) for all u, and I want to recreate the original function *f*(x)?

- Well, sum it up for every u:

$$f(x) = \int_{-\infty}^{+\infty} F(u)\left[\cos(2\pi ux) + i\sin(2\pi ux)\right]du$$

Discrete Fourier Transform

- Problem: an image is not an analogue signal that we can integrate.
- Therefore for $0 \leq x < N$ and $0 \leq u < N/2$:

$$F(u) = \sum_{x=0}^{N-1} f(x)\left[\cos\left(\frac{2\pi ux}{N}\right) - i\sin\left(\frac{2\pi ux}{N}\right)\right]$$

And the discrete inverse transform is:

$$f(x) = \frac{1}{N}\sum_{x=0}^{N-1} F(u)\left[\cos\left(\frac{2\pi ux}{N}\right) + i\sin\left(\frac{2\pi ux}{N}\right)\right]$$

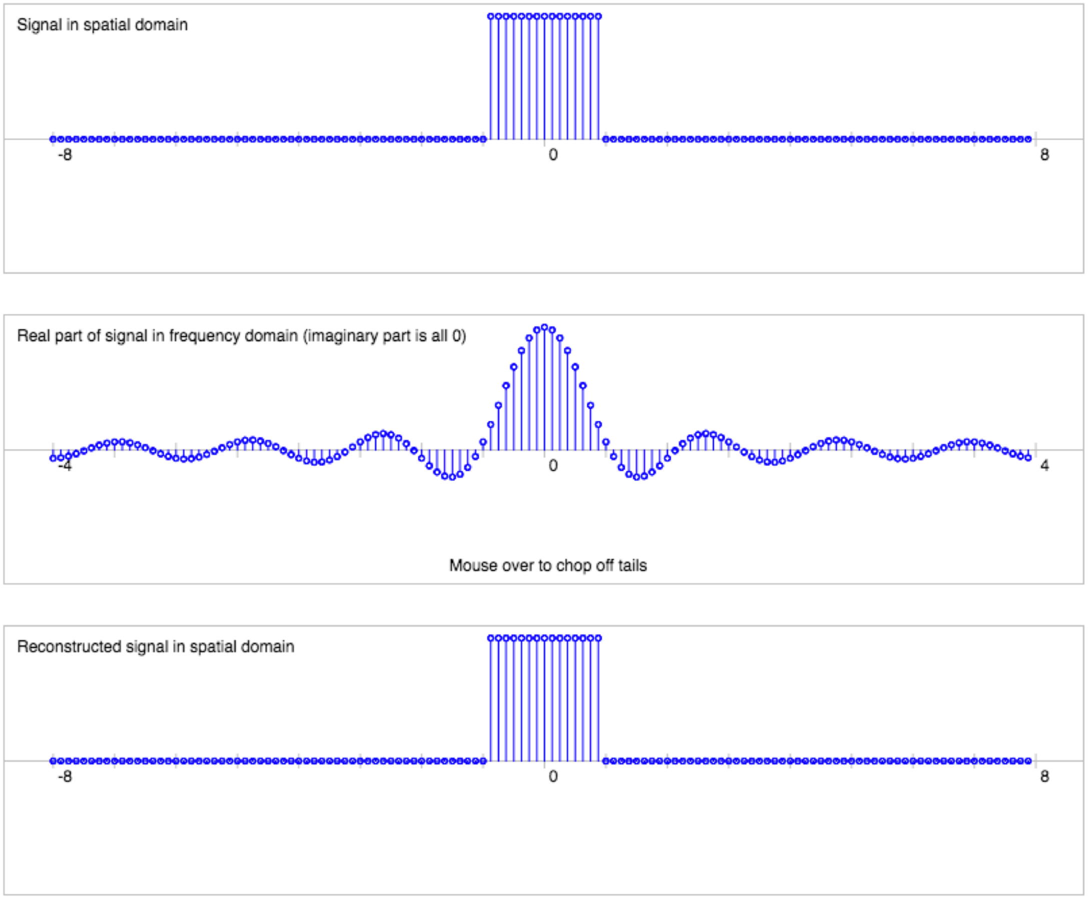
Discrete vs. Continuous

- Summation replaces integration
- Division by N (the number of discrete samples) makes the unit of repetition 1.
- For any signal (continuous or discrete)
  - f(x) is called the <u>spatial domain</u>
  - F(u) is called the <u>frequency domain</u>

Spatial vs. Frequency

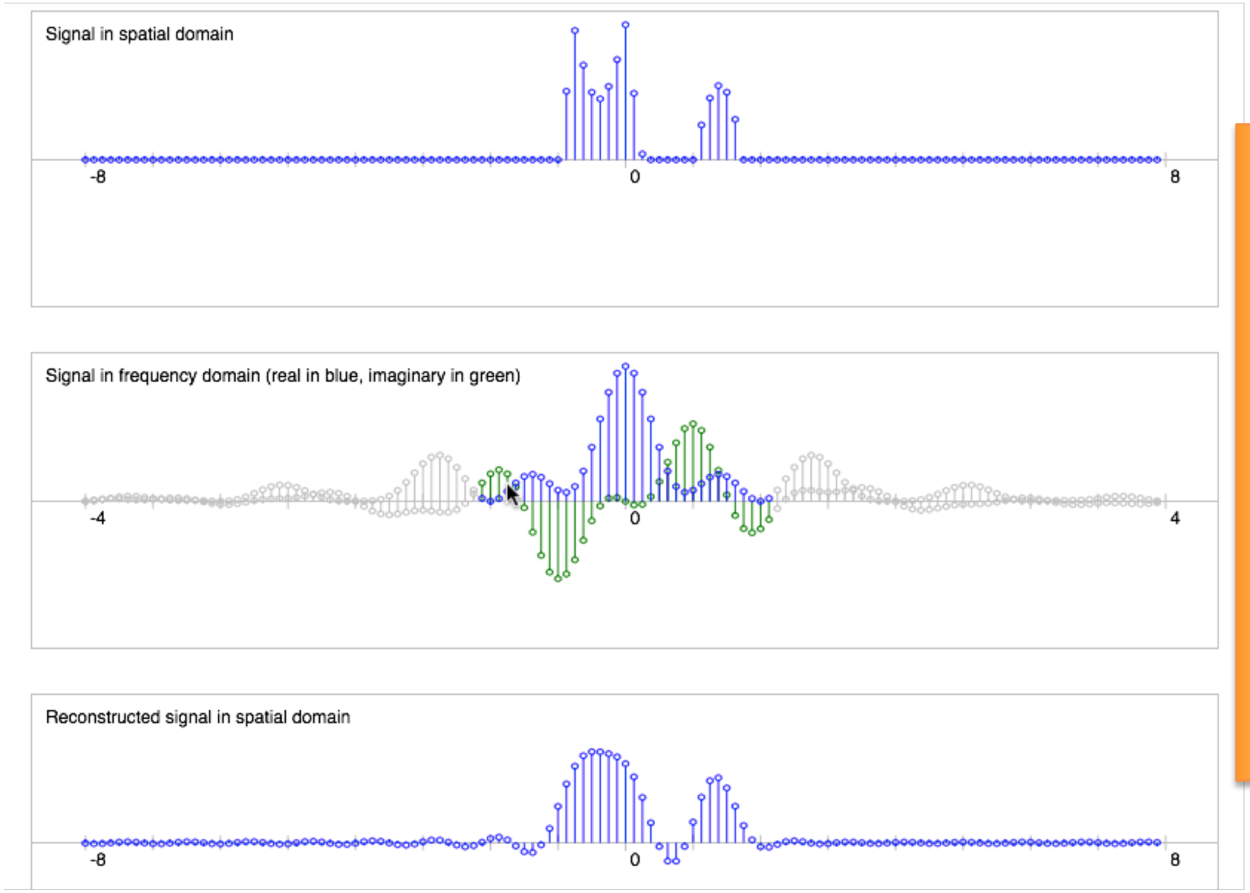
- Spatial domain representation size?
  - Given N samples, it is size N
- Frequency domain representation size?
  - A total of N/2 frequencies
    - Often plotted from –N/2 to N/2, but half are redundant
  - A complex number (2 values) per frequency
- The DFT is invertible, so the two representations are equivalent:
  - Exact same information and same size
  - The FFT is O(n log n)

Key Intuition: invertibility



http://madebyevan.com/dft/

An important intuition: "trimming" creates artifacts



http://madebyevan.com/dft/

An intuition for later…

Any frequency of the discrete Fourier transform

$$F(u) = \sum_{x=0}^{N-1} f(x)\left[cos\left({}^{2\pi ux}/_{N}\right) - isin\left({}^{2\pi ux}/_{N}\right)\right]$$

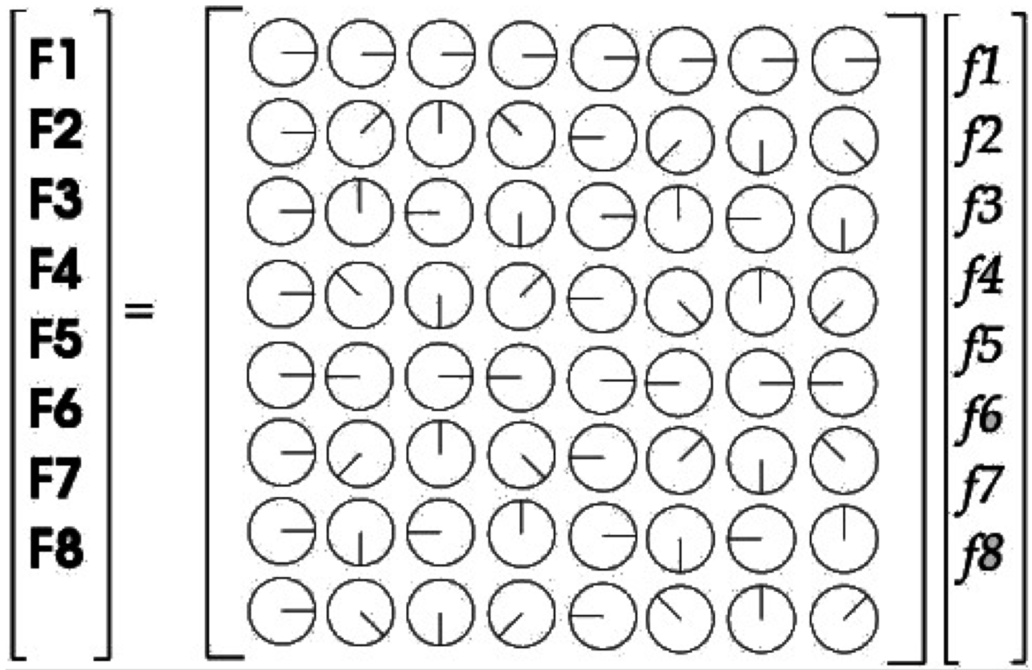
Can be rewritten as a complex dot product:

$$F(u) = \left[cos\left({}^{2\pi u0}/_{N}\right) - isin\left({}^{2\pi u0}/_{N}\right), \cdots, \right] \cdot \begin{bmatrix} f(0) \\ \vdots \\ f(N-1) \end{bmatrix}$$

So the Fourier transform is linear

An intuition … (cont.)

The full transform is a matrix equation



Each circle represents a complex number; x is cosine, y is sine

https://en.wikibooks.org/wiki/Digital_Signal_Processing/Discrete_Fourier_Transform

An intuition … (part 3)

- This matrix has special properties
  - Every row is orthogonal to every other
  - Every row has length $\sqrt{N}$
  - <u>It is a rotation and a scale of image space</u>
  - The inverse Fourier counter-rotates and counter scales

2D Fourier Transform

- So far, we have looked only at 1D signals
- For 2D signals, the continuous generalization is:

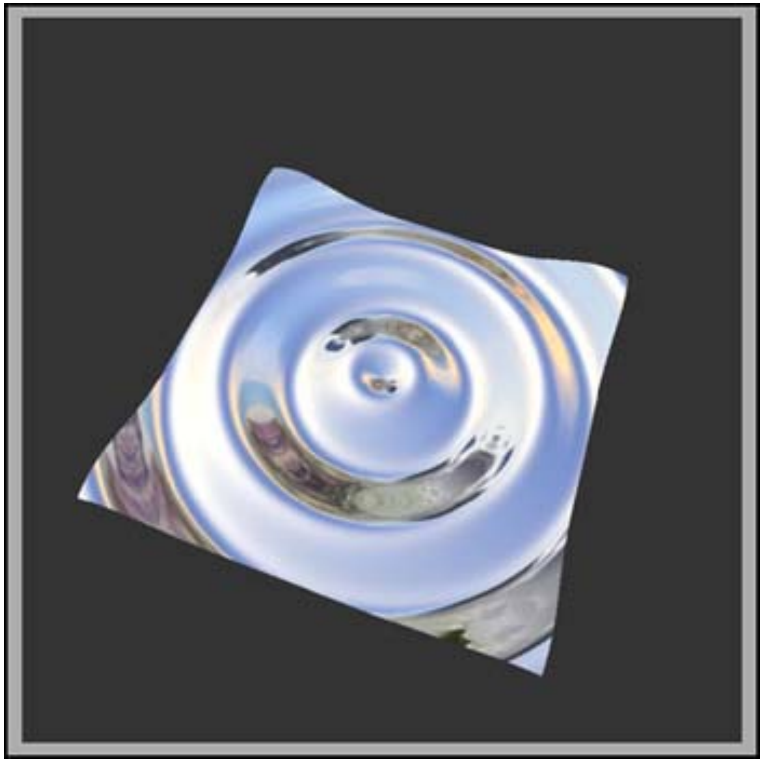
$$F(u,v) \equiv \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\left[\cos\left(2\pi(ux+vy)\right) - i\sin\left(2\pi(ux+vy)\right)\right]$$

- Note that frequencies are now two-dimensional
  - u= freq in x, v = freq in y
- Every frequency (u,v) has a real and an imaginary component.

Colorado State University

2D sine waves

- This looks like you'd expect in 2D



➢ Note that the frequencies don't have to be equal in the two dimensions.

http://images.google.com/imgres?imgurl=http://developer.nvidia.com/dev_content/cg/cg_examples/images/sine_wave_perturbation_ogl.jpg&imgrefurl=http://developer.nvidia.com/object/cg_effects_explained.html&usg=__0FimoxuhWMm59cbwhch0TLwGpQM=&h=350&w=350&sz=13&hl=en&start=8&sig2=dBEtH0hp5I1BExgkXAe_kg&tbnid=fcyrIaatfp0P3M:&tbnh=120&tbnw=120&ei=llCYSbLNL4miMoOwoP8L&prev=/images%3Fq%3D2D%2Bsine%2Bwave%26gbv%3D2%26hl%3Den%26sa%3DG

2D Discrete Fourier Transform

$$F(u,v)=\sum_{x=-N/2}^{N/2}\sum_{y=-N/2}^{N/2} f(x,y)\left[\cos\left(\frac{2\pi}{N}(ux+vy)\right)-i\sin\left(\frac{2\pi}{N}(ux+vy)\right)\right]$$

- What happened to the bounds on x & y?
- How big is the discrete 2D frequency space representation?

2D Frequency Space

- Remember that:
  - Cosine is an even function: cos(x) = cos(-x)
  - Sine is an odd function: sin(x) = -sin(-x)
- So
  - $F(u,v) = a+ib \Rightarrow F(-u, -v) = a-ib$
- And
  - $F(-u,v) = a+ib \Rightarrow F(u, -v) = a-ib$
- But
  - $F(u,v) = a+ib \Rightarrow F(-u, v) = ???$

2D Frequency Space (cont)

- Size of 2D Frequency representation:
  - One dimension must vary from –N/2 to N/2, while the other varies from 0 to N/2
    - Doesn't matter which is which
  - N * (N/2) * 2 values per frequency = N^2
  - Same as the source spatial representation

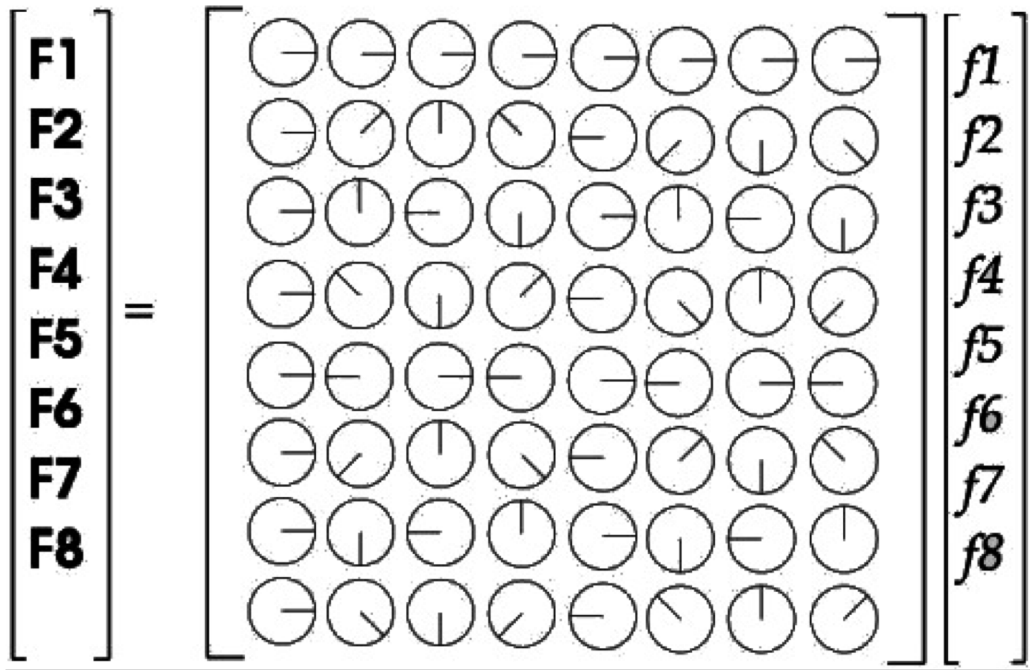
Colorado State University

Showing Frequency Space

- To display a frequency space:
  - We plot it from –N/2 to N/2 in both dimensions
  - The result is symmetric about the origin (and therefore redundant)
  - We can't plot a complex number, so we show the magnitude at every pixel sqrt($a^2 + b^2$)
    - Thus discarding the phase information
    - Phase plots are also possible ($\tan^{-1}(b/a)$)

Transform as Linear Operation

The full transform is a matrix equation



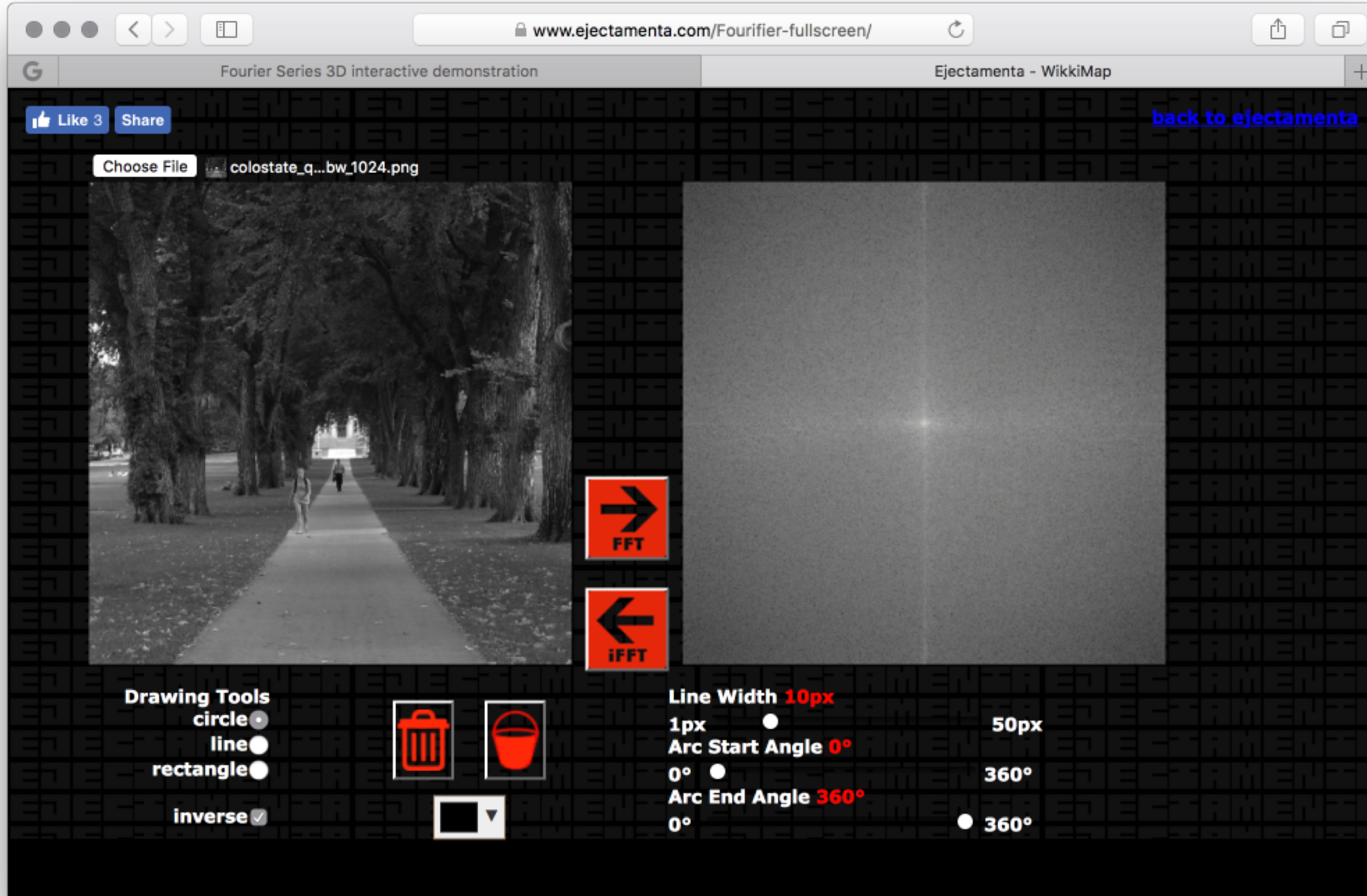
Each circle represents a complex number; x is cosine, y is sine

https://en.wikibooks.org/wiki/Digital_Signal_Processing/Discrete_Fourier_Transform

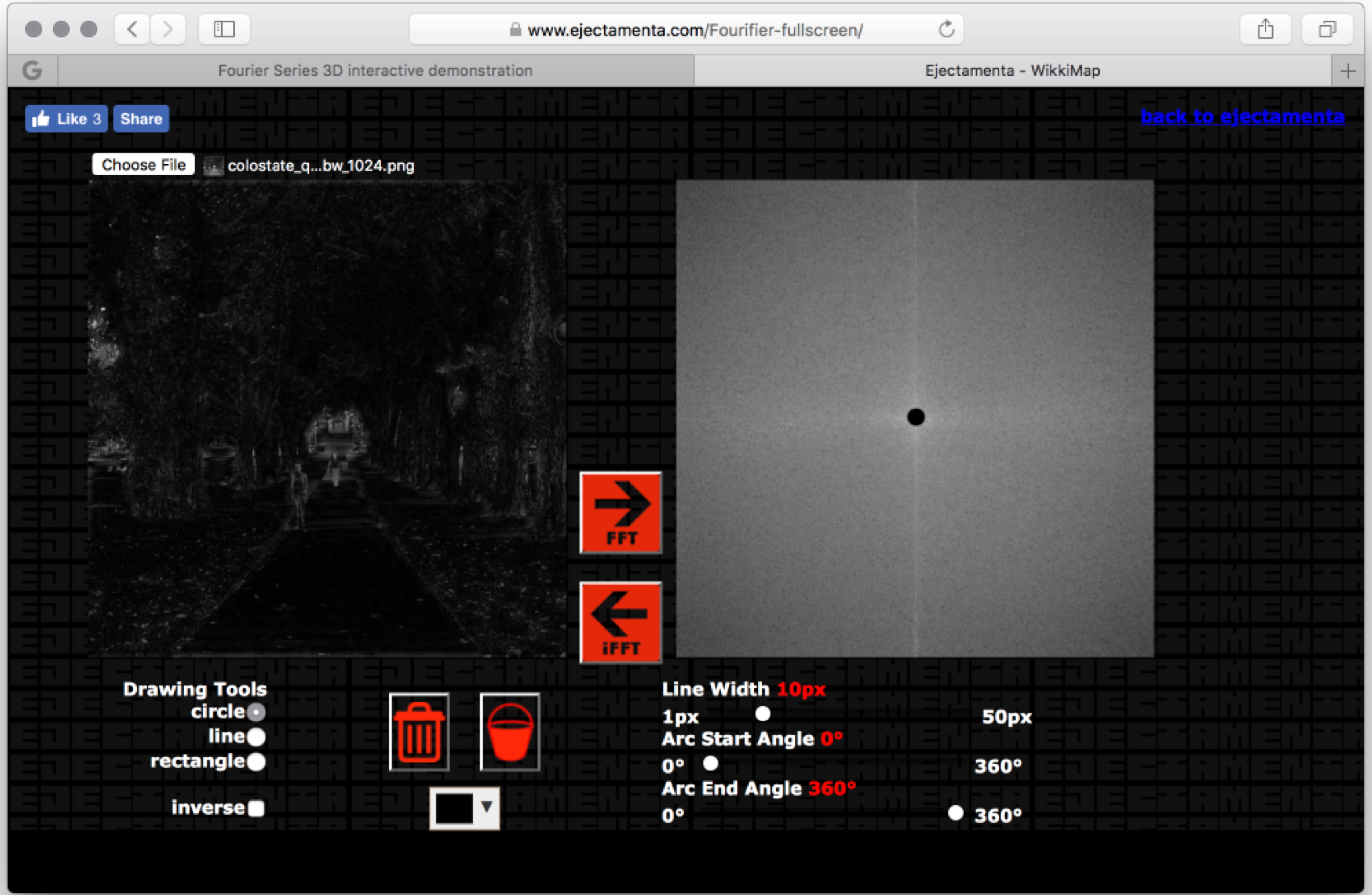
Fourier as Linear Transform

(N+2) x 1 (N+2) x 2N 2N x 1

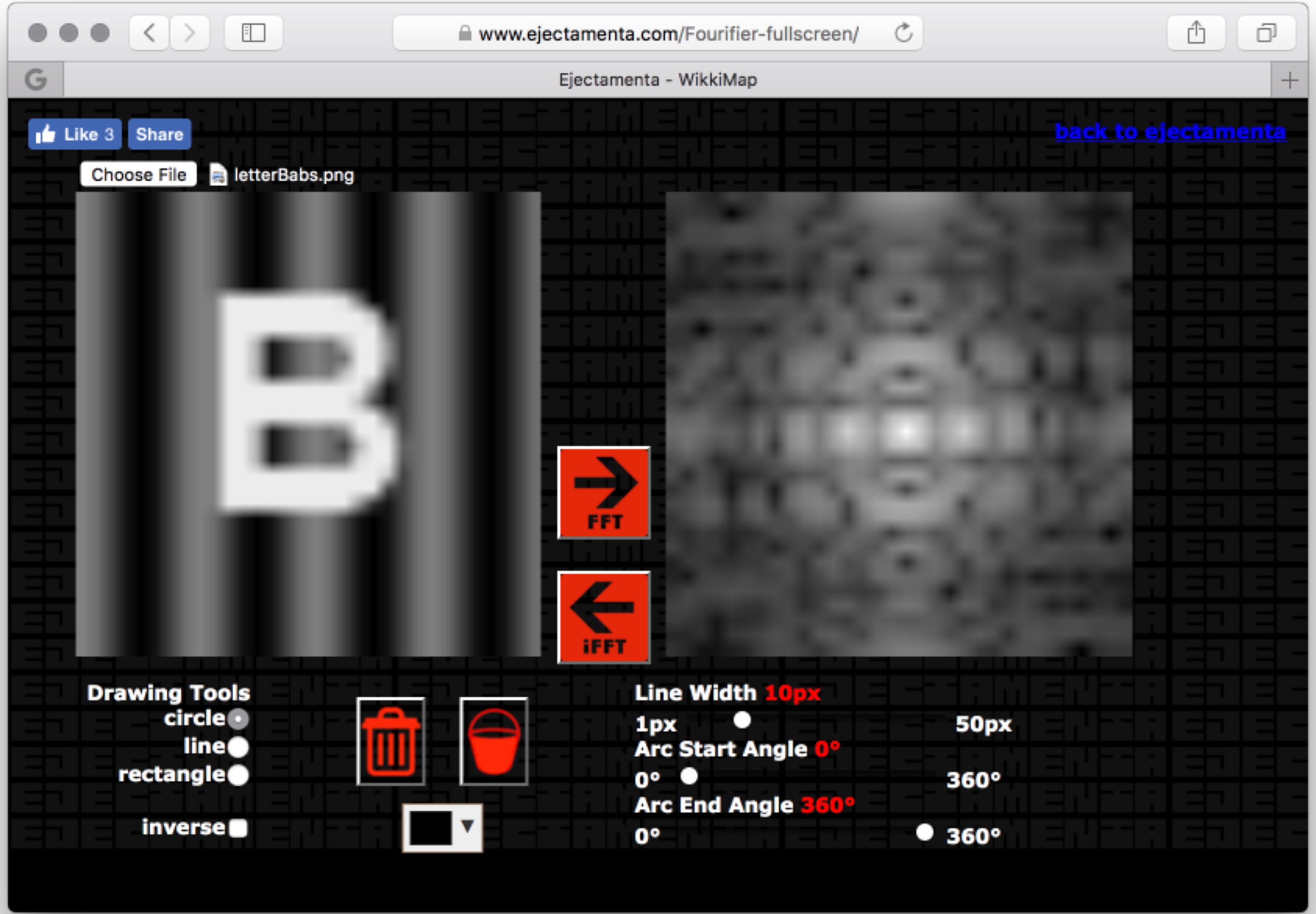
$$\begin{bmatrix} R(F(0)) \\ I(F(0)) \\ R(F(1)) \\ I(F(1)) \\ \vdots \\ R\left(F\left(N/2\right)\right) \\ I\left(F\left(N/2\right)\right) \end{bmatrix} = \begin{bmatrix} \cos(0) & 0 & \cos(0) & 0 & \cdots & \cos(0) & 0 \\ 0 & \sin(0) & 0 & \sin(0) & \cdots & 0 & \sin(0) \\ \cos\left(\frac{2\pi\cdot 0}{N}\right) & 0 & \cos\left(\frac{2\pi\cdot 1}{N}\right) & 0 & \cdots & \cos\left(\frac{2\pi\cdot(N-1)}{N}\right) & 0 \\ 0 & \sin\left(\frac{2\pi\cdot 0}{N}\right) & 0 & \sin\left(\frac{2\pi\cdot 1}{N}\right) & \cdots & 0 & \sin\left(\frac{2\pi\cdot(N-1)}{N}\right) \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ \cos\left(\frac{2\pi\left(\frac{N}{2}\right)\cdot 0}{N}\right) & 0 & \cos\left(\frac{2\pi\left(\frac{N}{2}\right)\cdot 1}{N}\right) & 0 & \cdots & \cos\left(\frac{2\pi\left(\frac{N}{2}\right)\cdot(N-1)}{N}\right) & 0 \\ 0 & \sin\left(\frac{2\pi\left(\frac{N}{2}\right)\cdot 0}{N}\right) & 0 & \sin\left(\frac{2\pi\left(\frac{N}{2}\right)\cdot 1}{N}\right) & \cdots & 0 & \sin\left(\frac{2\pi\left(\frac{N}{2}\right)\cdot(N-1)}{N}\right) \end{bmatrix} \begin{bmatrix} I(0) \\ I(0) \\ I(1) \\ I(1) \\ \vdots \\ I(N-1) \\ I(N-1) \end{bmatrix}$$

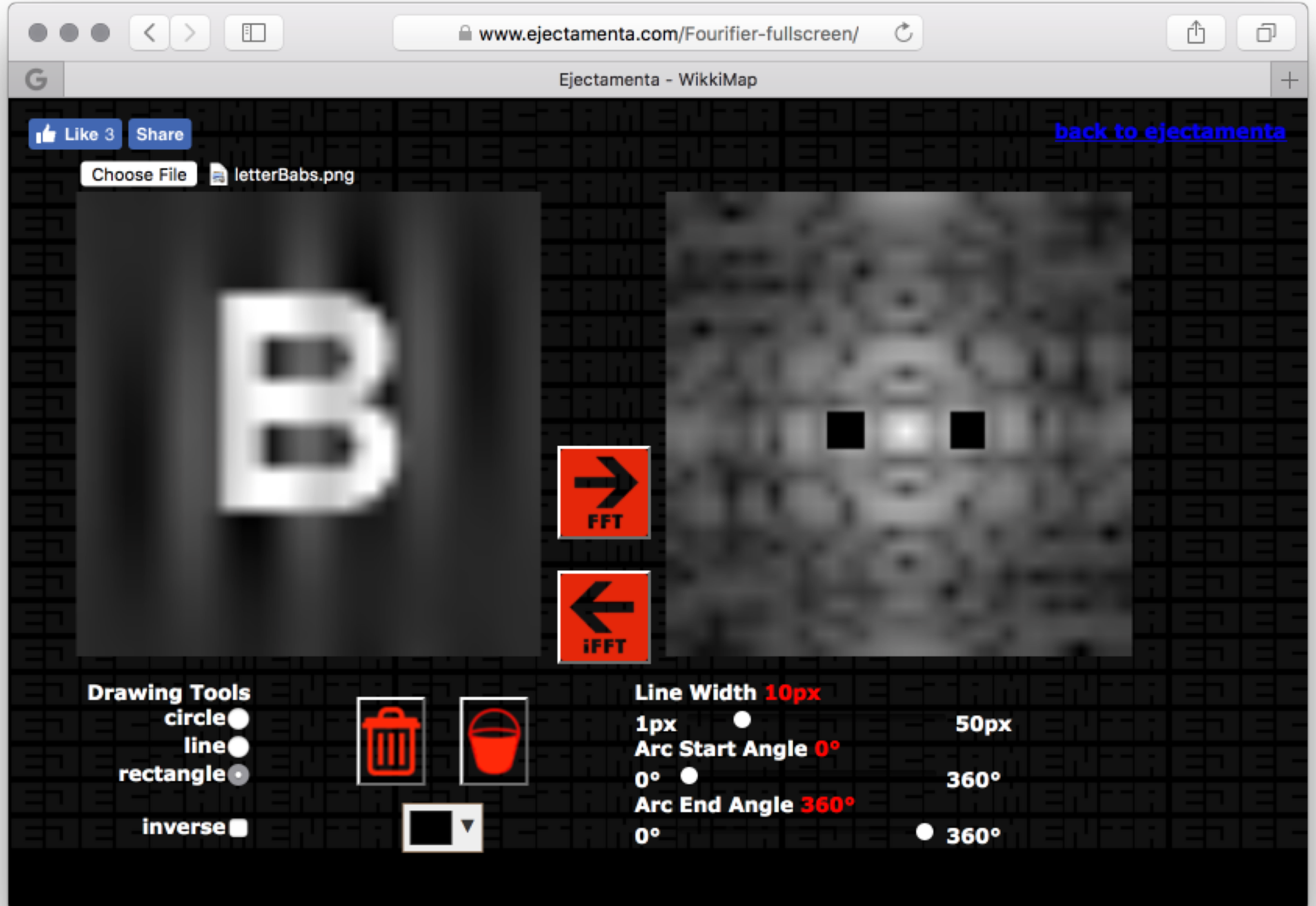


Colorado State University



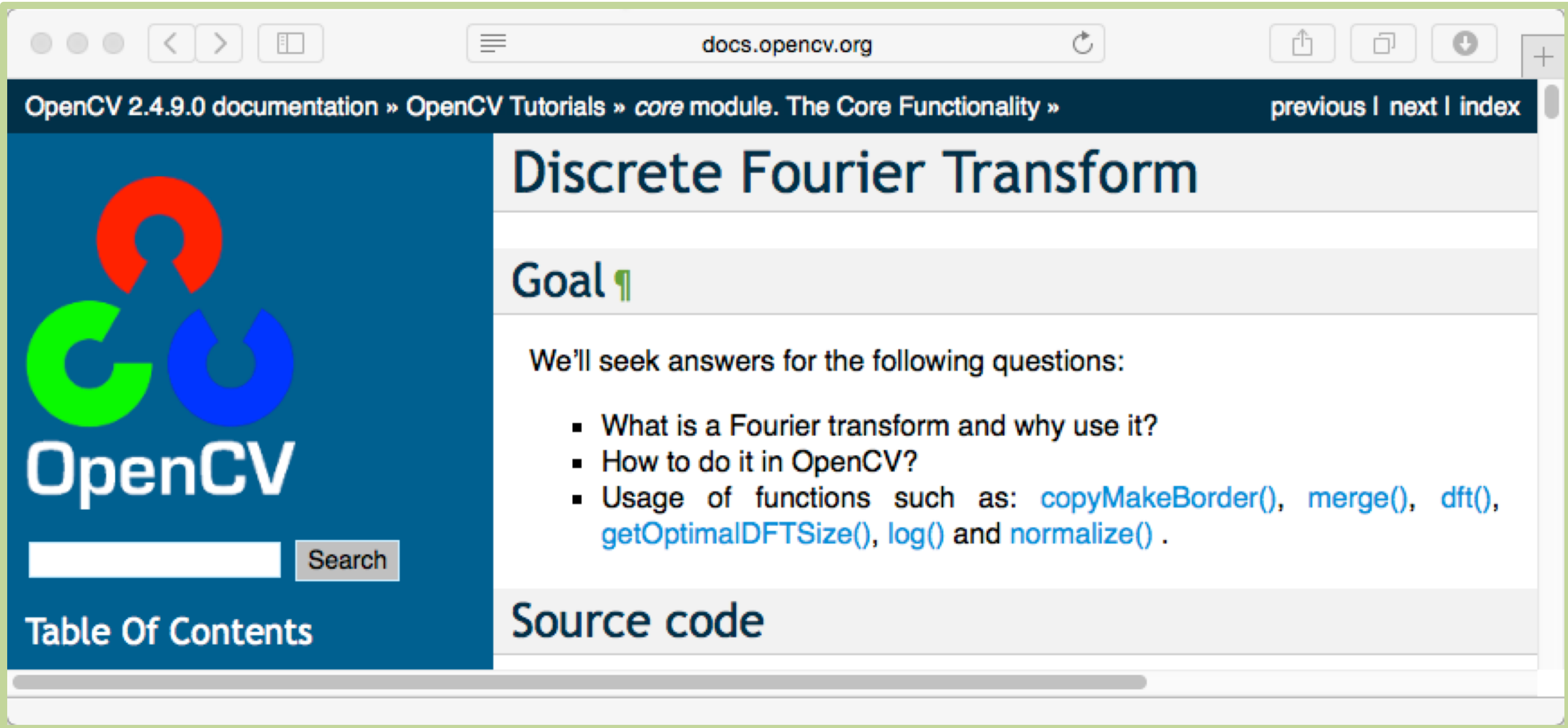
Colorado State University


www.ejectamenta.com/Fourifier-fullscreen/
Ejectamenta - WikkiMap
Like 3
Share
back to ejectamenta
Choose File
letterBabs.png
FFT
iFFT
Drawing Tools
circle
line
rectangle
inverse
Line Width 10px
1px
50px
Arc Start Angle 0°
0°
360°
Arc End Angle 360°
0°
360°



Colorado State University

In OpenCV – Fourier Tutorial


docs.opencv.org

OpenCV 2.4.9.0 documentation » OpenCV Tutorials » *core* module. The Core Functionality » previous | next | index

OpenCV

Search

Table Of Contents

Discrete Fourier Transform

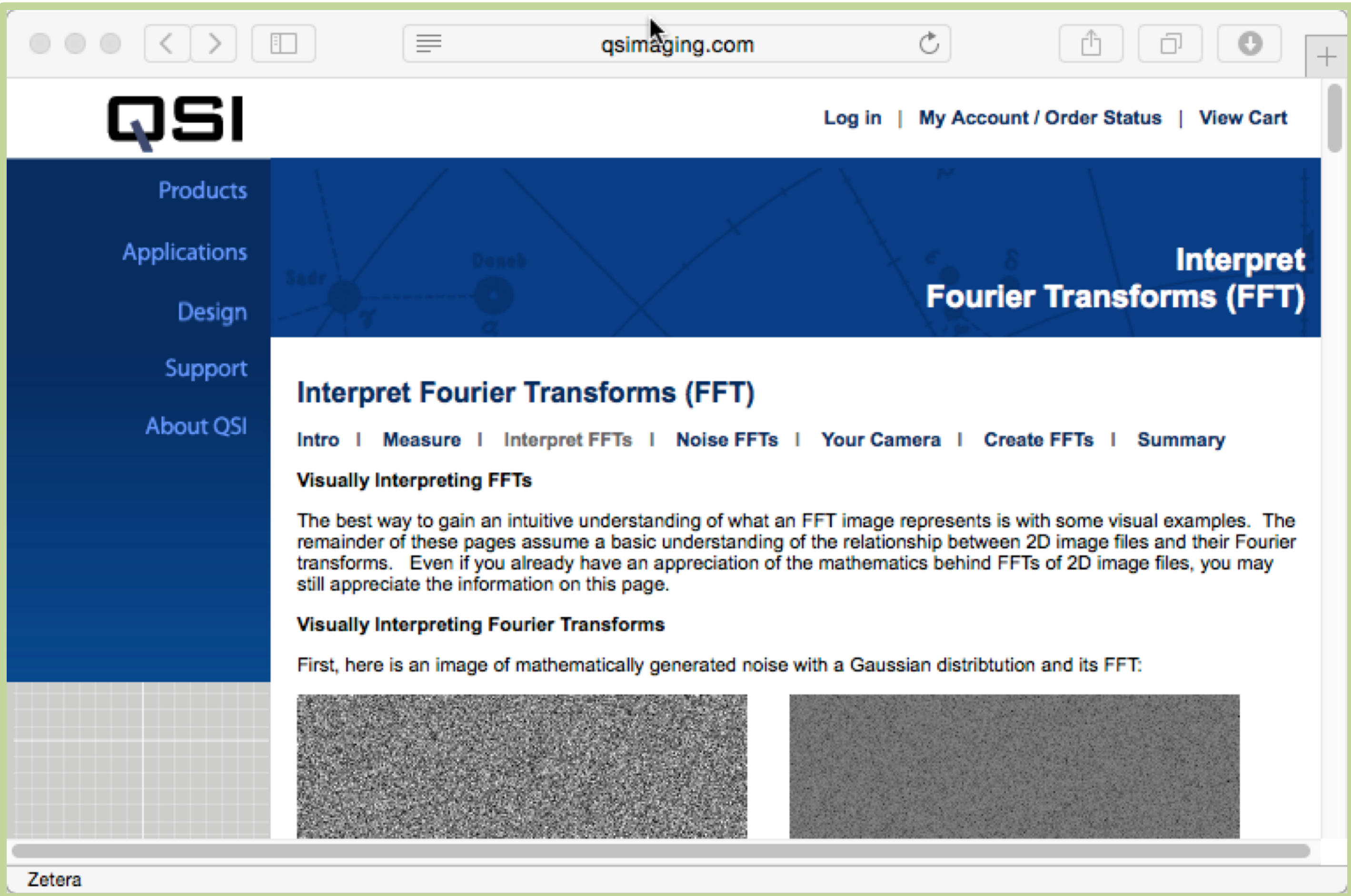
Goal

We'll seek answers for the following questions:

- What is a Fourier transform and why use it?
- How to do it in OpenCV?
- Usage of functions such as: copyMakeBorder(), merge(), dft(), getOptimalDFTSize(), log() and normalize() .

Source code

Helpful Images and Examples



Colorado State University

Non-local properties

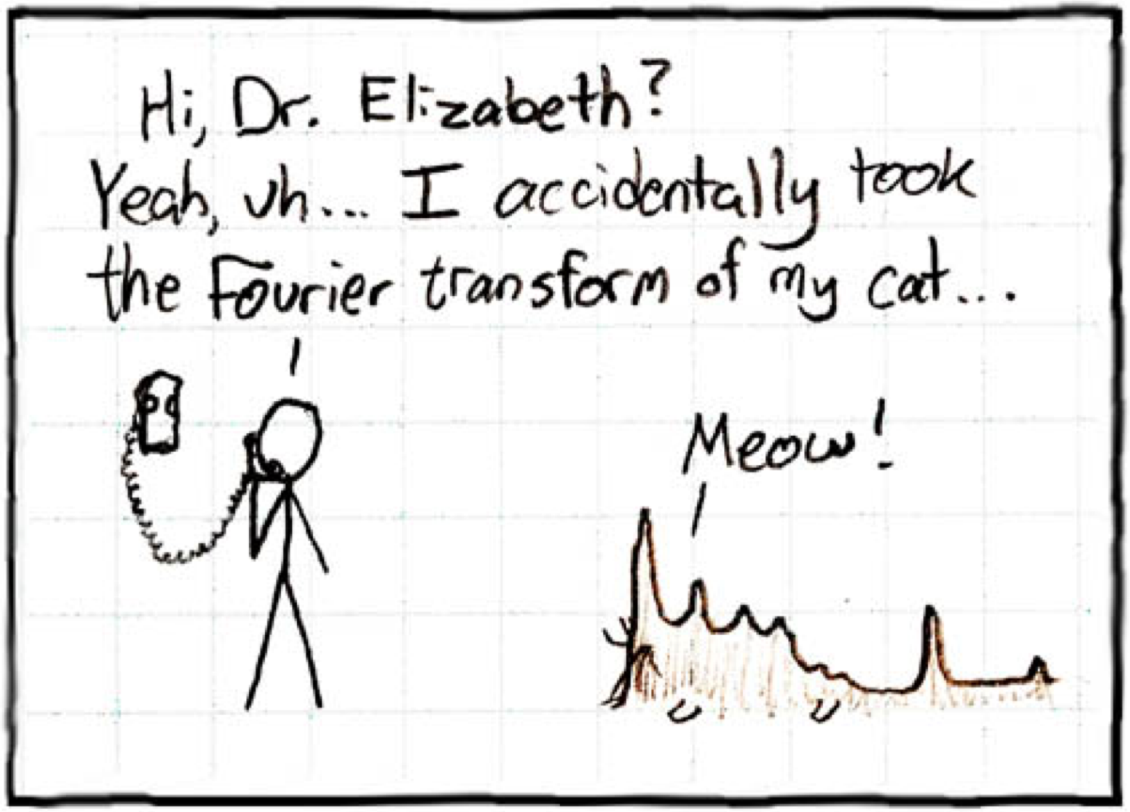
- Change one spatial pixel in an image, *every frequency value changes*

- Change one value in the frequency domain, *every spatial pixel changes.*

- Frequencies describe the image as a whole, not useful for describing part of an image

XKCD

https://xkcd.com/26/

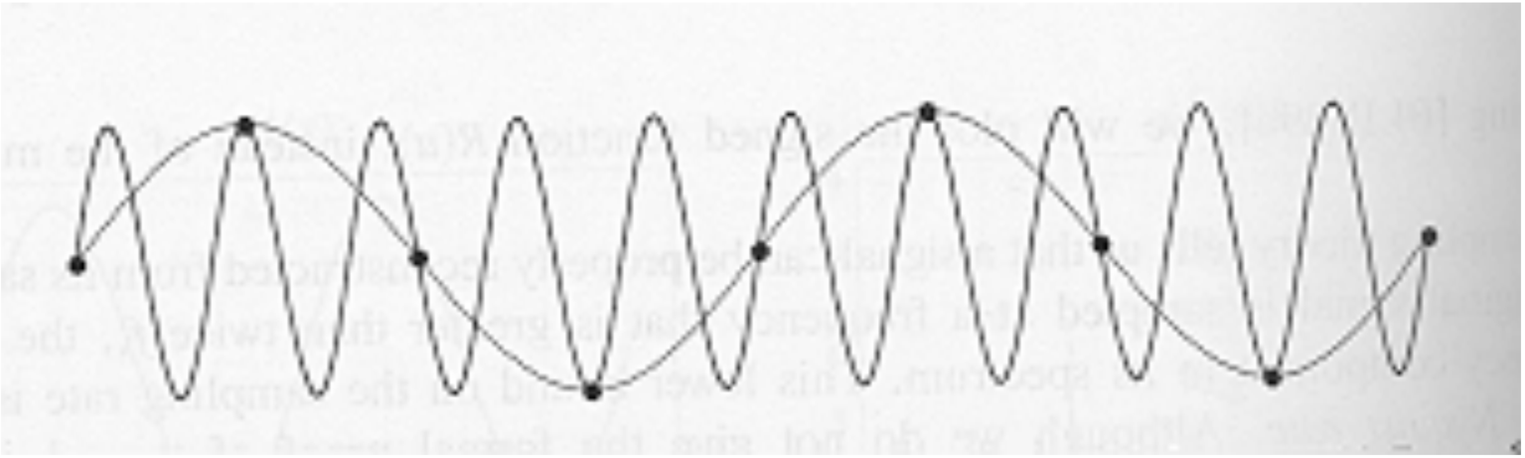


Review: Why Study Fourier?

- Relates continuous to discrete
  - Continuous: underlying signal
  - Discrete: sampling of signal
- Tells us how much information is lost
  - Total energy in continuous frequencies above N/2
- Explains aliasing

The Nyquist Rate

- What if the frequency is above N/2?
  - You have fewer than one sample per half-cycle
  - High frequencies look like lower frequencies



Graphic from *"Computer Graphics:Principles and Practice" by Foley, van Dam, Feiner & Hughes.*

Aliasing – Another View