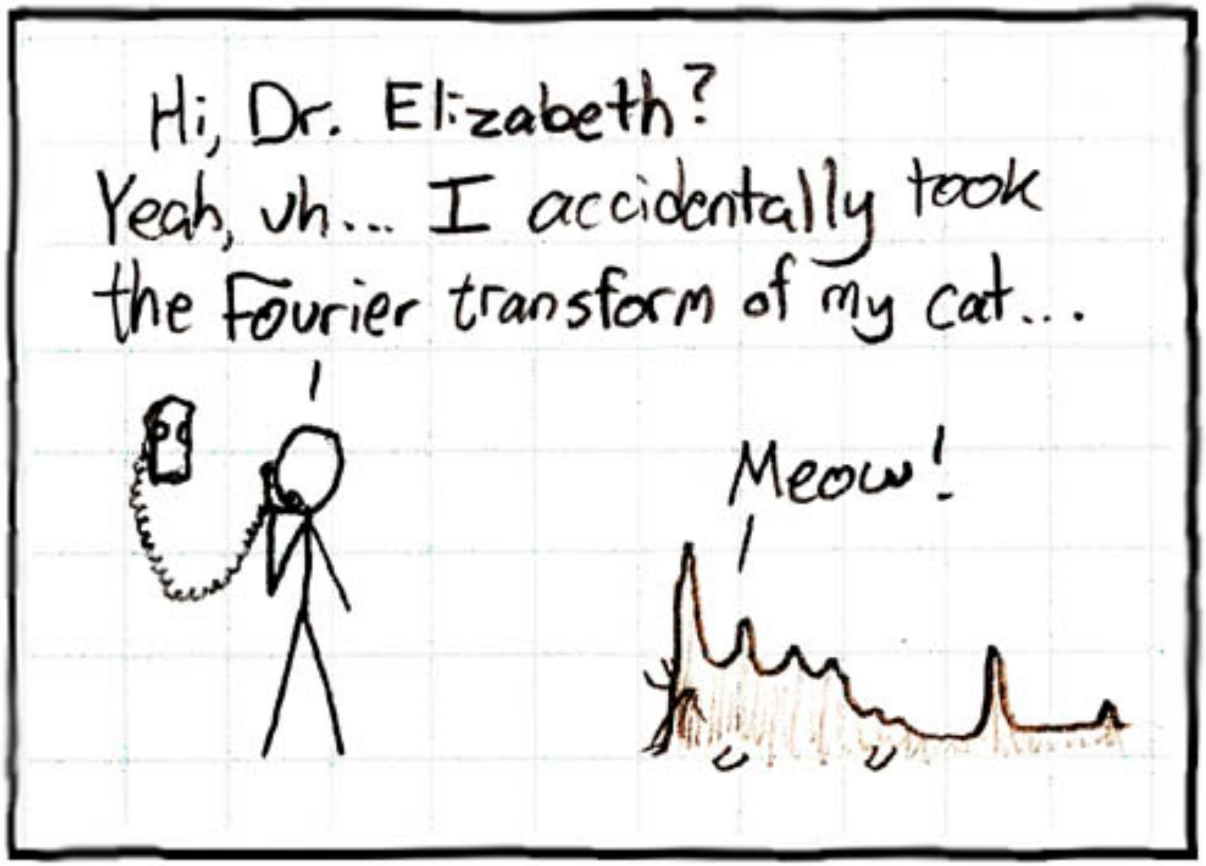
6 – Fourier Optics

- See any 'harmonics' in this? You will get what I am saying by the end of this lecture…
- One more image on next slide too :)

6 – Fourier Optics

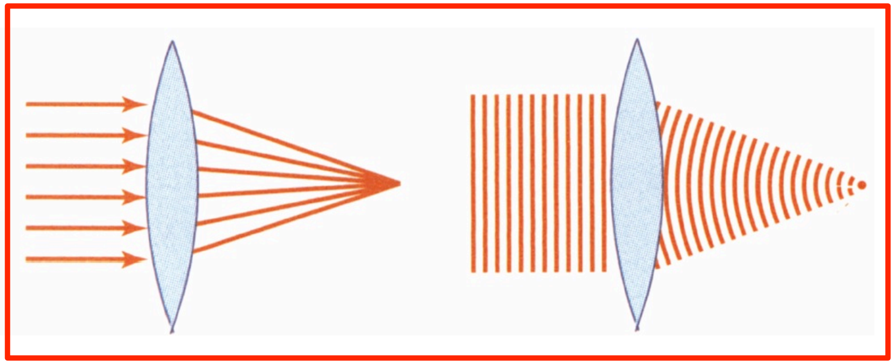

Hi, Dr. Elizabeth?
Yeah, uh... I accidentally took the Fourier transform of my cat...
Meow!

- This week will be the most 'real' and maybe 'fun' that the Fourier transform has ever been for you :)

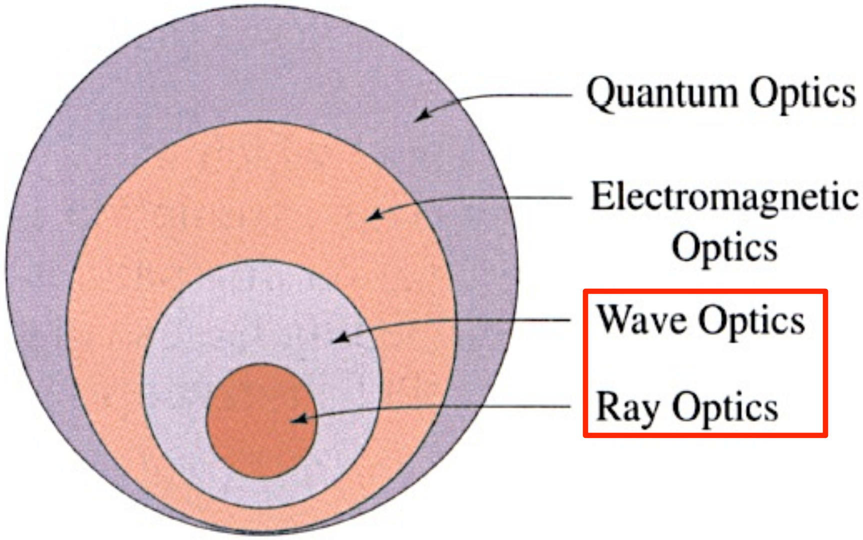
http://groups.csail.mit.edu/netmit/sFFT/images/fourier.jpg

Today…

- Today, we will mainly use a bit of ray and wave optics, but it will be far more complex than that and hard to visualize (need to rely on the mathematics).



Credit: Fund. Photonics – Fig. 2.3-1



Credit: Fund. Photonics – Fig. 1.0-1

- Topics:

Some of the figures today are from CH4 of Fund. of Photonics or wiki.

(1) Fourier Transform (mathematical)
(2) Example Application to Diffraction from a Slit (mathematical and optical)
(3) Fourier Transform and Lab this Week (optical)
(4) Example Application to Image (qualitative)

- Fourier Transform *… transforms a function function in the time domain y(t) or spatial domain y(z), into the frequency domain y(w) or y(f) (w=2πf). And the inverse as well!*

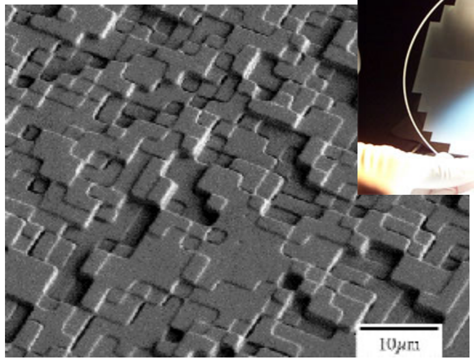
$$y(f)=\int_{-\infty}^{+\infty} y(x)\times e^{-i2\pi f x}\,dx \quad\Leftrightarrow\quad y(x)=\int_{-\infty}^{+\infty} y(f)\times e^{i2\pi f x}\,df$$

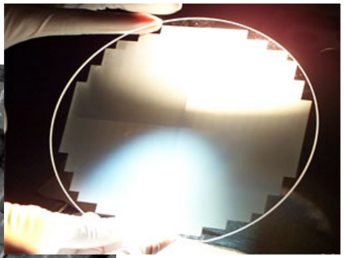
- The harmonics (sines, cosines) are introduced by Euler's formula:

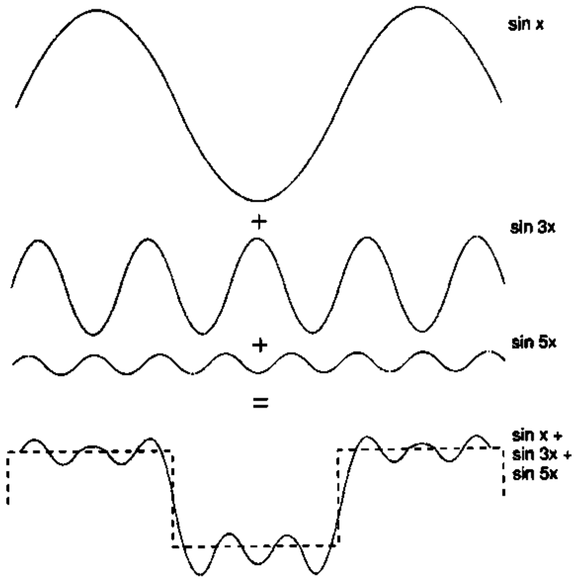
$$e^{ix}=\cos x+i\sin x$$
$$e^{-ix}=\cos x-i\sin x$$

Were you still wondering how they calculated the pattern needed for the holographic gratings we tested?

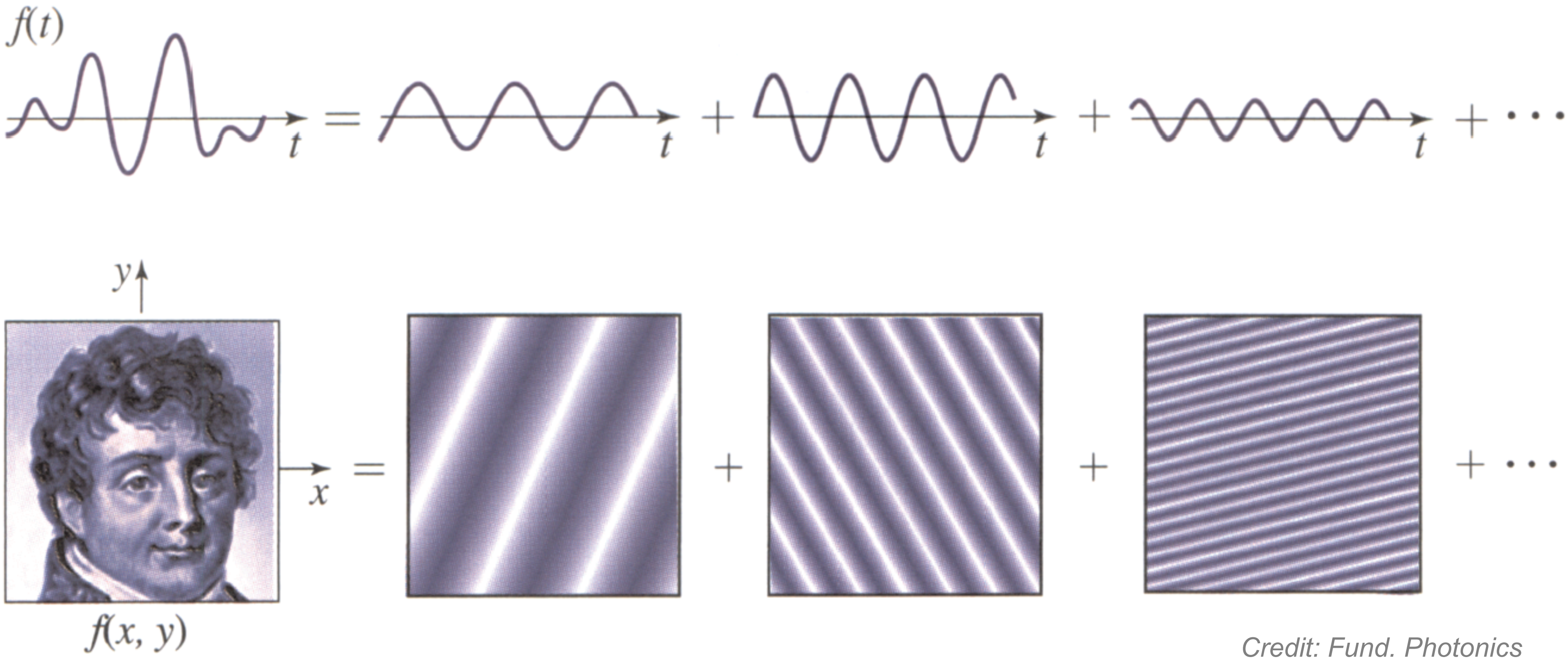
The answer soon!







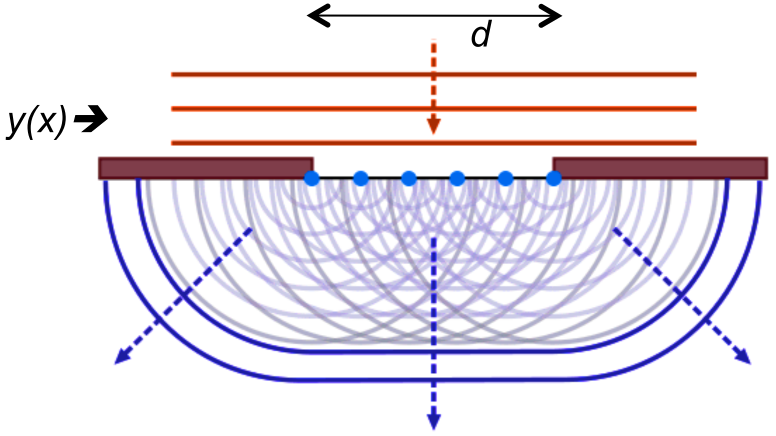
- Any waveform, image, etc.. can be expressed as the superposition of numerous harmonics… the more harmonics you are willing to have, the closer the representation of the original waveform/ image (sound like quality level in jpeg or mpeg compression?).



Credit: Fund. Photonics

Single Slit Example

- Consider the *irradiance (I, W/m²)* pattern for the single slit diffraction experiment… easiest function, y(x)



$$y(x) = 0 \ for \ |x| > d/2$$

$$y(x) = 1 \ for \ |x| < d/2$$

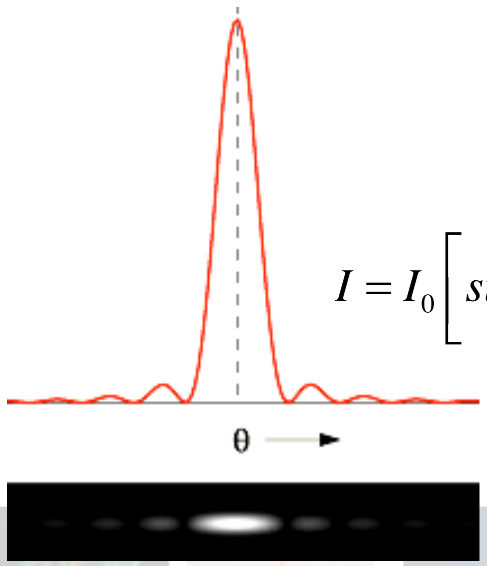
$$E_{\max}(k) = \int_{-d/2}^{+d/2} 1 \times e^{-i2\pi f x} \, dx$$

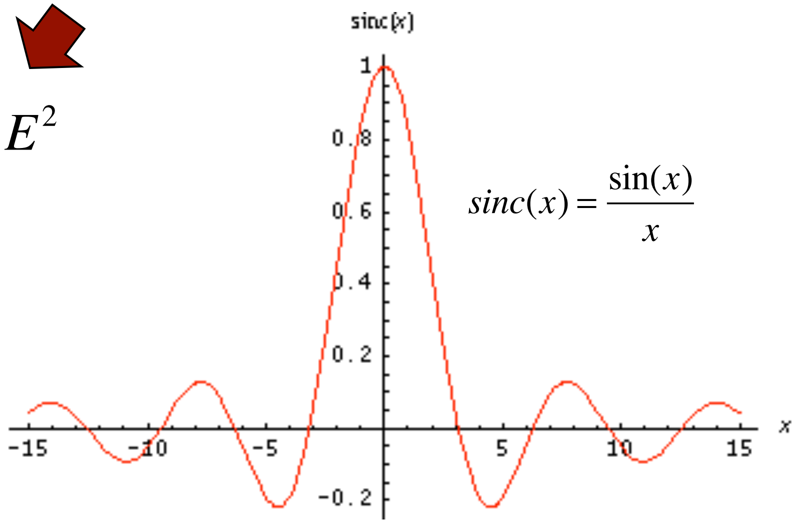
$$E_{\max}(k) = d \ sinc\left(\frac{\pi d}{\lambda}\sin\theta\right)$$

… Fourier: time (s) to frequency (f=1/s) domain
… Fourier: distance (m) to wavenumber (k=2π/λ, 1/m)

$$I = E^2$$

$$I = I_0\left[sinc\left(\frac{\pi d}{\lambda}\sin\theta\right)\right]^2 (W/m^2)$$





$$sinc(x) = \frac{\sin(x)}{x}$$

Rectangular Slit Example

- Can do the same for rectangular slit (Ex. in Fund. Photonics), only difference is 2D…

$$\boxed{E_{\max}(k) = \int_{-D_y/2}^{+D_x/2} \int_{-D_y/2}^{+D_y/2} 1 \times e^{-i2\pi(f_x x + f_y y)}\, dxdy}$$

EXERCISE 4.3-1

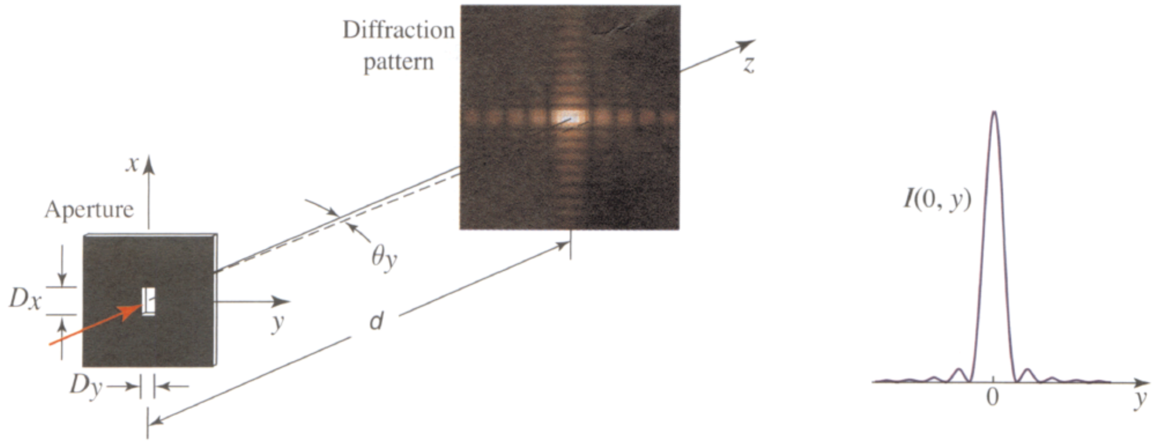
Fraunhofer Diffraction from a Rectangular Aperture. Verify that the Fraunhofer diffraction pattern from a rectangular aperture, of height and width D_x and D_y respectively, observed at a distance d is

$$I(x,y) = I_o \operatorname{sinc}^2 \boxed{\frac{D_x x}{\lambda d}} \operatorname{sinc}^2 \boxed{\frac{D_y y}{\lambda d}} \qquad (4.3\text{-}6)$$

where $I_o(D_x D_y/\lambda d)^2 I_i$ is the peak intensity and $\operatorname{sinc}(x) = \sin(\pi x)/(\pi x)$. Verify that the first zeros of this pattern occur at $x = \pm\lambda d/D_x$ and $y = \pm\lambda d/D_y$, so that the angular divergence of the diffracted light is given by

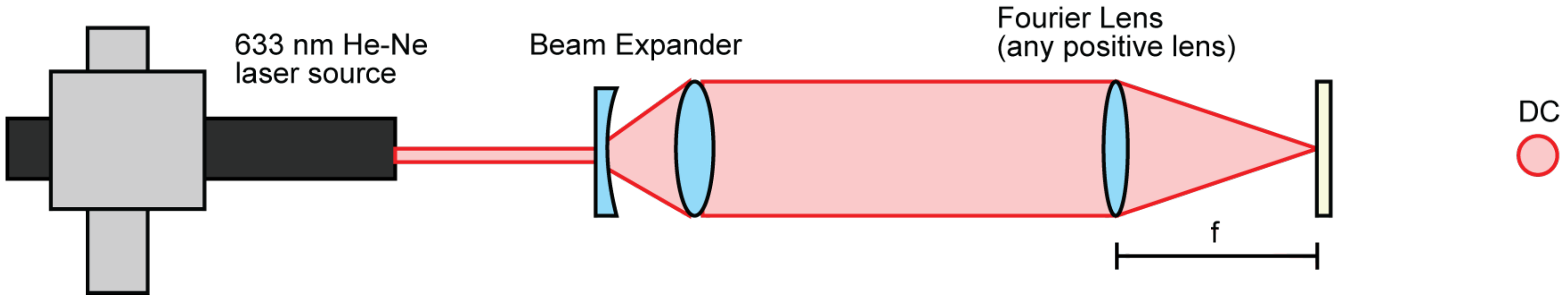
$$\boxed{\theta_x = \frac{\lambda}{D_x}, \qquad \theta_y = \frac{\lambda}{D_y}.} \qquad (4.3\text{-}7)$$

If $Dy < D_x$, the diffraction pattern is wider in the y direction than in the x direction, as illustrated in Fig. 4.3-3.

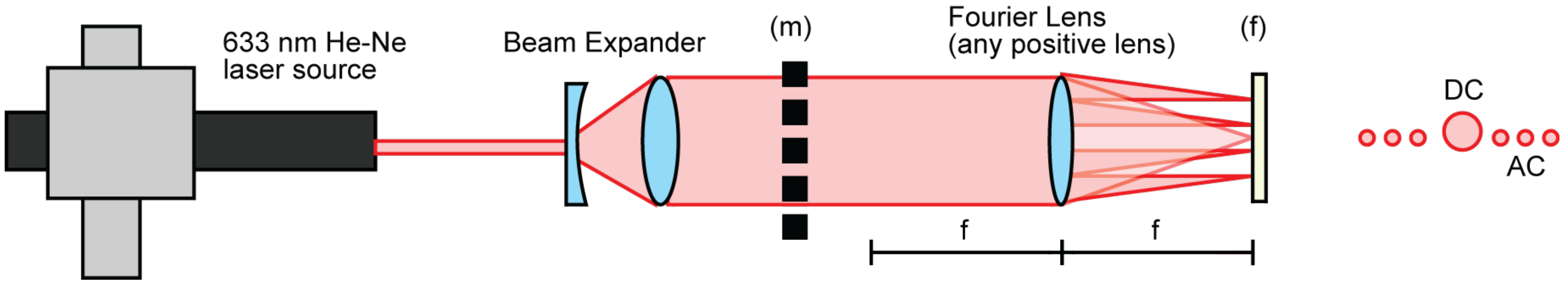


How to do with Optics…

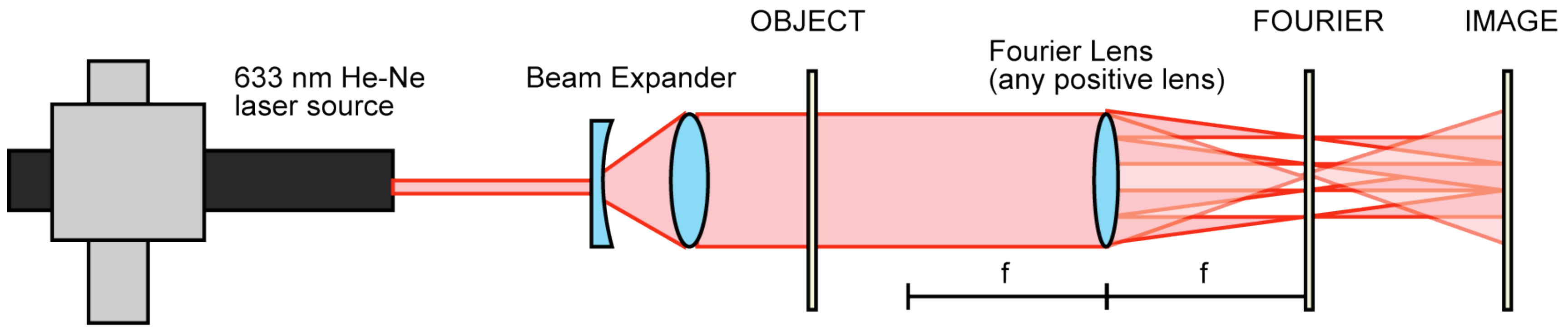
- Light coming from infinity (parallel rays) onto a lens all converges to the focal point of a lens, the beam is broad and unchanging so only 'DC' component (no harmonics or 'AC' frequencies)…



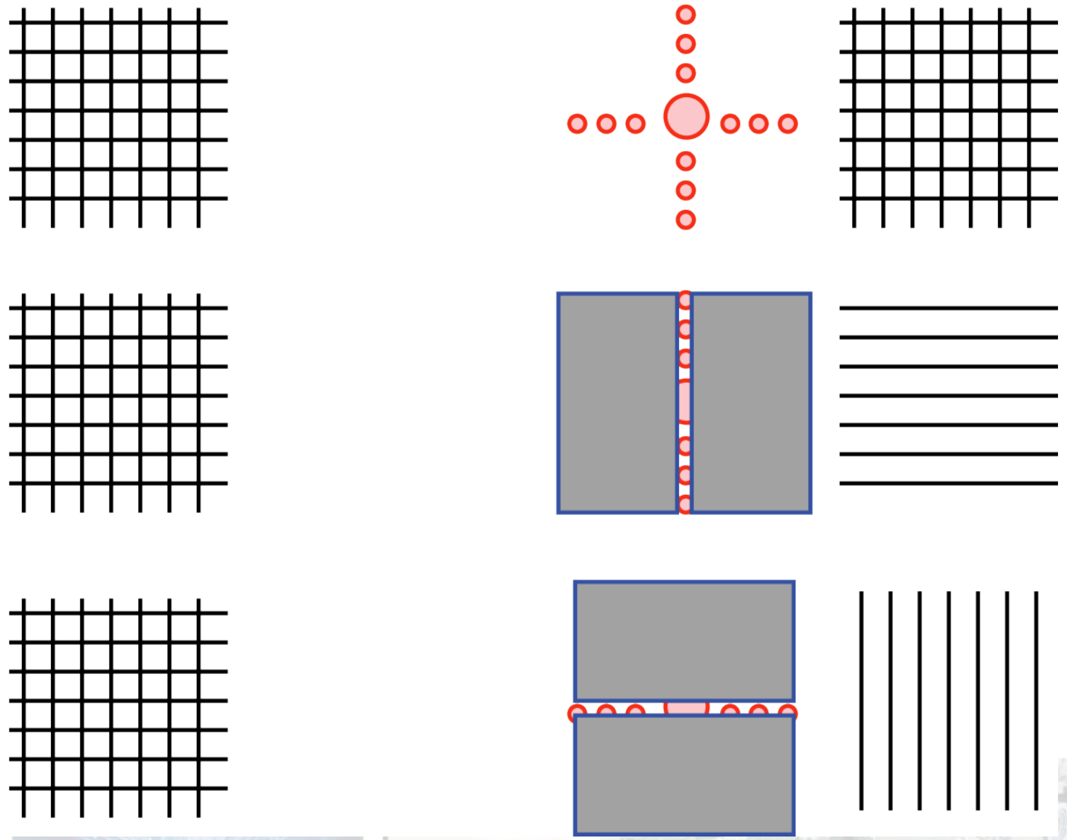
- What if we have an object with sharp edges? Diffraction! Projected to discrete angles…
  … *a surface that changes rapidly has 'higher frequencies' that comprise it.*



Filter Images at Light Speed!



- This week you will filter AC and DC components of a wire mesh! See at right…
- *So why do we get the cross dot pattern in the Fourier plane? Why the big bright central dot?*
- *Now, add a vertical filter, why only horizontal lines are left in the final image?*
- *Same question for horizontal filter…*
- *Could we use to clean a dirty laser beam (dust, scratches, etc..)?*



Review! Take a break!

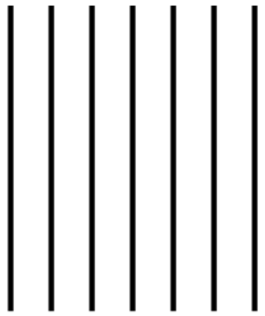
- The Fourier transform can be:

(a) used to predict diffraction patterns.
(b) implemented at the focal point of a simple lens at the speed of light.
(c) be utilized to 'clean up' imperfections created on a laser beam by sharp features such as scratches on lenses and mirrors, etc..
(d) all the above.

- If you use the vertical wires shown at right, what would you see in the Fourier plane? *Hint, think of how light would diffract off the wires…*

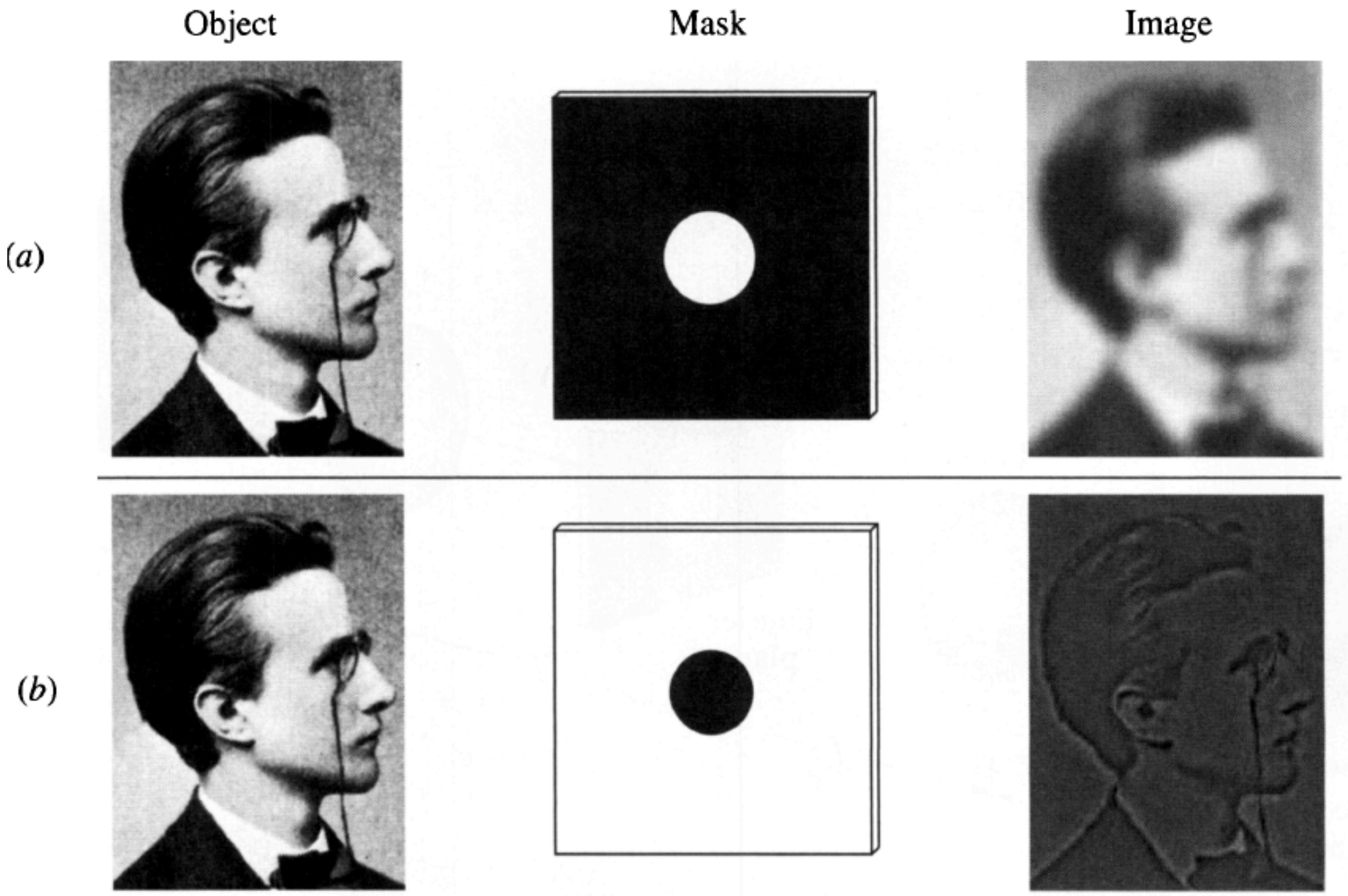
(a) Vertical and horizontal 1D arrays of dots.
(b) Vertical 1D array of dots.
(c) Horizontal 1D array of dots.
(d) Straight lines, like that shown at right, but larger (magnified) due to the lens.

- Whew! That's enough. Lets take a break!



You Can Do Complex Images Too!

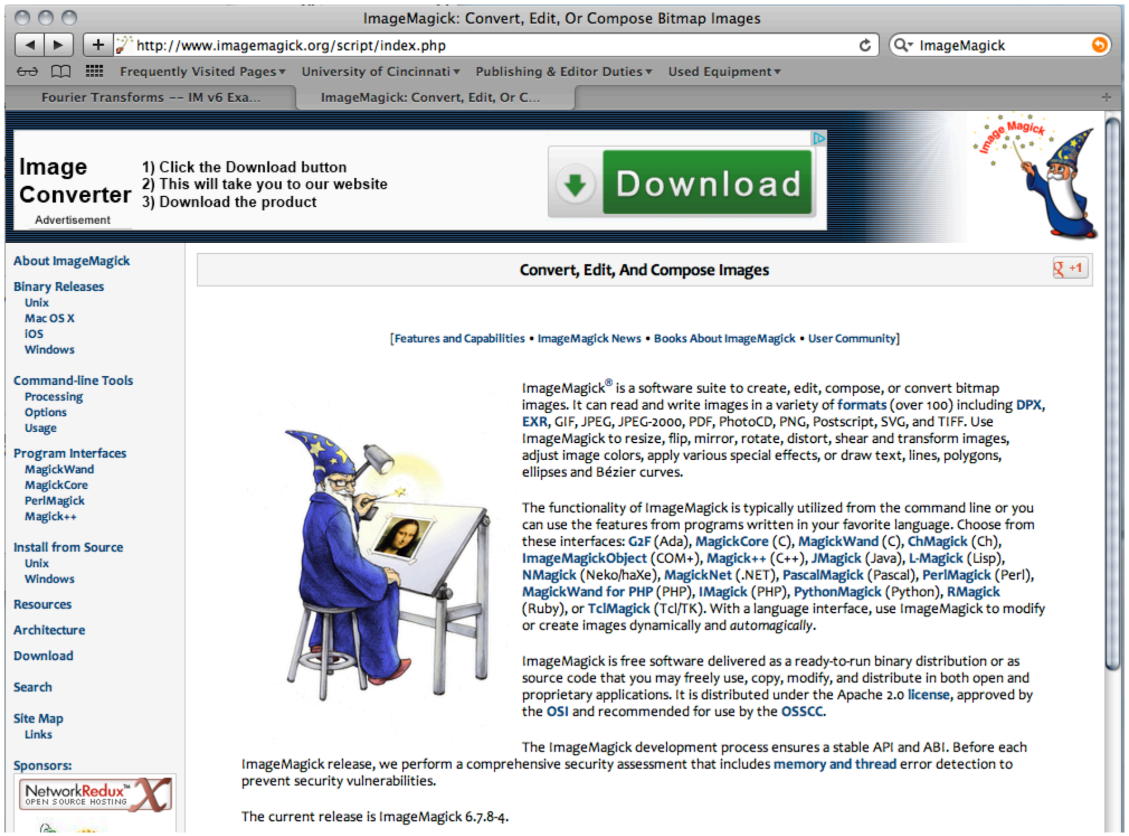
- So how do these examples work? (from Fund. of Photonics CH4)



Example FFT on Images

- Some really cool 'fast Fourier transform' or FFT examples (and credit for next few slides) can be found here:

http://www.imagemagick.org/Usage/fourier/



ImageMagick: Convert, Edit, Or Compose Bitmap Images

http://www.imagemagick.org/script/index.php

Frequently Visited Pages ▾ University of Cincinnati ▾ Publishing & Editor Duties ▾ Used Equipment ▾

Fourier Transforms -- IM v6 Exa... ImageMagick: Convert, Edit, Or C...

Image Converter
1) Click the Download button
2) This will take you to our website
3) Download the product
Download
Advertisement

About ImageMagick
Binary Releases
Unix
Mac OS X
iOS
Windows
Command-line Tools
Processing
Options
Usage
Program Interfaces
MagickWand
MagickCore
PerlMagick
Magick++
Install from Source
Unix
Windows
Resources
Architecture
Download
Search
Site Map
Links
Sponsors:

Convert, Edit, And Compose Images

[Features and Capabilities • ImageMagick News • Books About ImageMagick • User Community]

ImageMagick® is a software suite to create, edit, compose, or convert bitmap images. It can read and write images in a variety of formats (over 100) including DPX, EXR, GIF, JPEG, JPEG-2000, PDF, PhotoCD, PNG, Postscript, SVG, and TIFF. Use ImageMagick to resize, flip, mirror, rotate, distort, shear and transform images, adjust image colors, apply various special effects, or draw text, lines, polygons, ellipses and Bézier curves.

The functionality of ImageMagick is typically utilized from the command line or you can use the features from programs written in your favorite language. Choose from these interfaces: G2F (Ada), MagickCore (C), MagickWand (C), ChMagick (Ch), ImageMagickObject (COM+), Magick++ (C++), JMagick (Java), L-Magick (Lisp), NMagick (Neko/haXe), MagickNet (.NET), PascalMagick (Pascal), PerlMagick (Perl), MagickWand for PHP (PHP), IMagick (PHP), PythonMagick (Python), RMagick (Ruby), or TclMagick (Tcl/TK). With a language interface, use ImageMagick to modify or create images dynamically and *automagically*.

ImageMagick is free software delivered as a ready-to-run binary distribution or as source code that you may freely use, copy, modify, and distribute in both open and proprietary applications. It is distributed under the Apache 2.0 license, approved by the OSI and recommended for use by the OSSCC.

The ImageMagick development process ensures a stable API and ABI. Before each ImageMagick release, we perform a comprehensive security assessment that includes memory and thread error detection to prevent security vulnerabilities.

The current release is ImageMagick 6.7.8-4.

- You can also do transforms in MATLAB (will be a homework problem this week).

Example FFT on Images

‣ Now, lets simply try a Fast Fourier Transform round trip on the Lena image. That is, we simply do the forward transform and immediately apply the inverse transform to get back the original image. Then we will compare the results to see the level of quality produced…

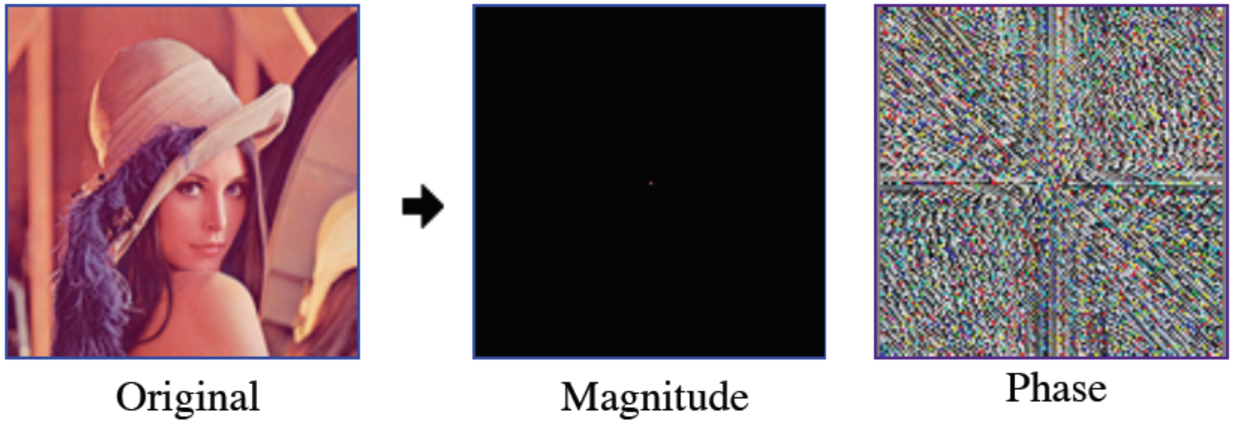
```
time convert lena.png -fft -ift lena_roundtrip.png

echo -n "RMSE = "
compare -metric RMSE lena.png lena_roundtrip.png null:
echo -n "PAE = "
compare -metric PAE  lena.png lena_roundtrip.png null:
```

‣ The images are only about 0.22% off.. Why? The software (like most) uses the 'Fast Fourier Transform'

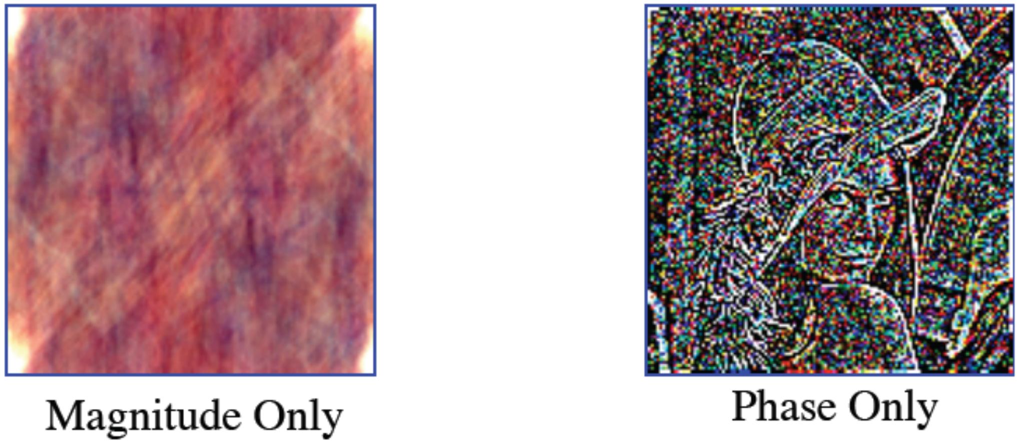
"By far the most commonly used FFT is the Cooley–Tukey *algorithm*. This is a divide and conquer algorithm that recursively breaks down a DFT of any composite size N = N1N2 into many smaller DFTs of sizes N1 and N2, along with O(N) multiplications by complex roots of unity traditionally called twiddle factors (after Gentleman and Sande, 1966)."

Example FFT on Images

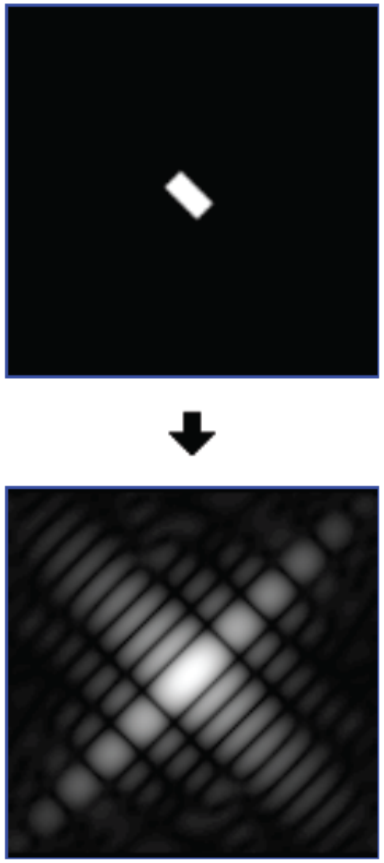
- Here are the transform images:



- If you restore just from magnitude or phase, here is what you get… for phase only the outlines make sense (why do they pop out and dominate?)



- Recognize this? Like diffraction big features become small, and small features become big…

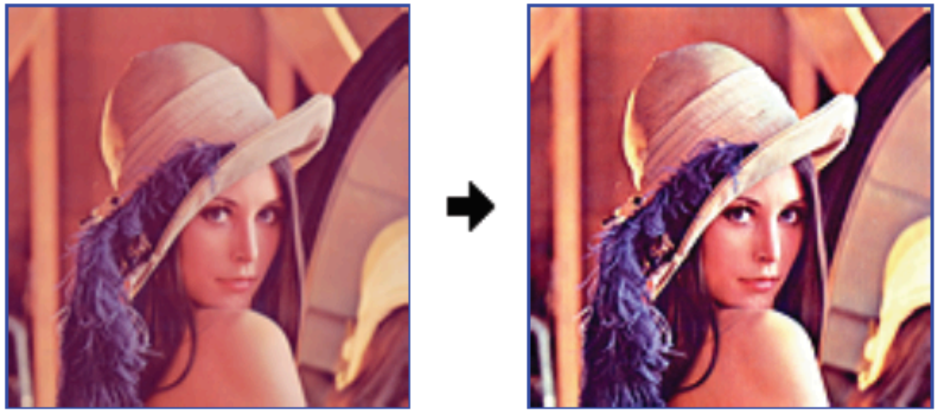


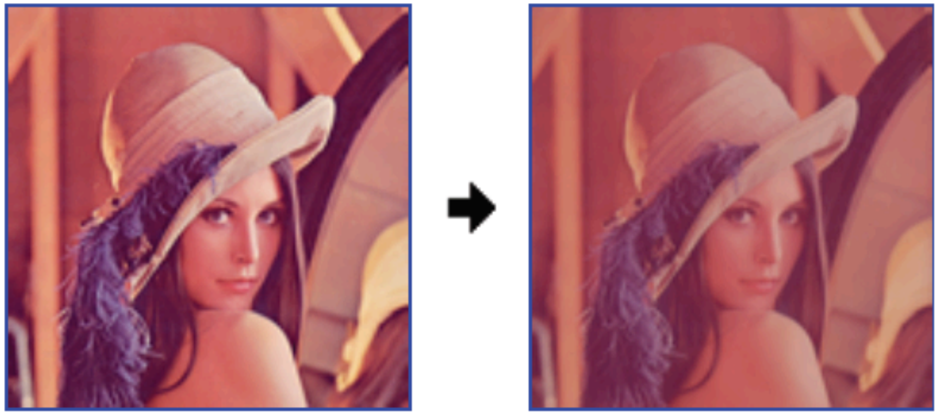
Example FFT on Images

- One can adjust the contrast in an image by performing the forward Fourier transform, raising the magnitude image to a power and then using that with the phase in the inverse Fourier transform.

- To increase, the contrast, one uses an exponent slightly less than one and to decrease the contrast, one uses an exponent slightly greater than one.

```
convert lena.png -fft \
        \( -clone 0 -evaluate pow 0.9 \) -delete 0 \
        +swap -ift lena_plus_contrast.png

convert lena.png -fft \
        \( -clone 0 -evaluate pow 1.1 \) -delete 0 \
        +swap -ift lena_minus_contrast.png
```

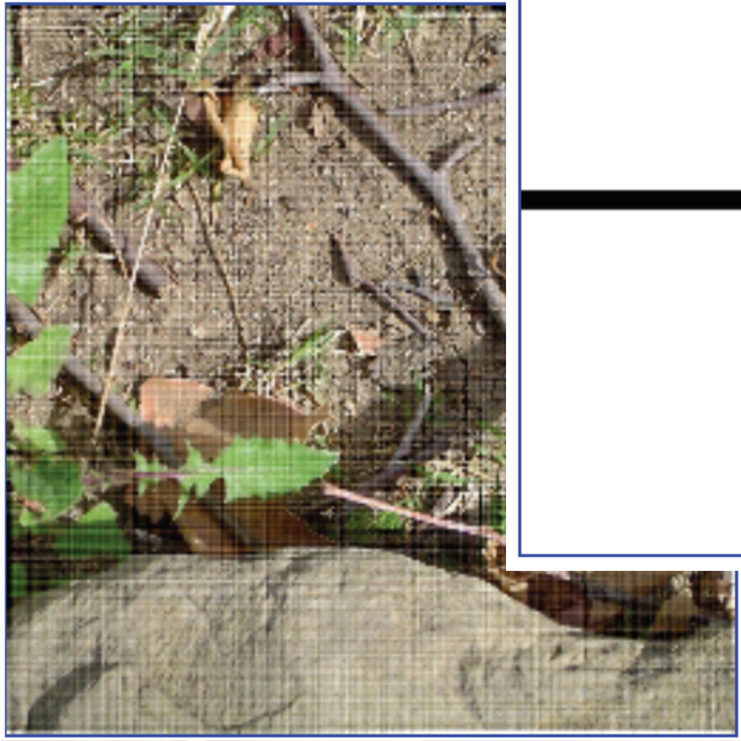




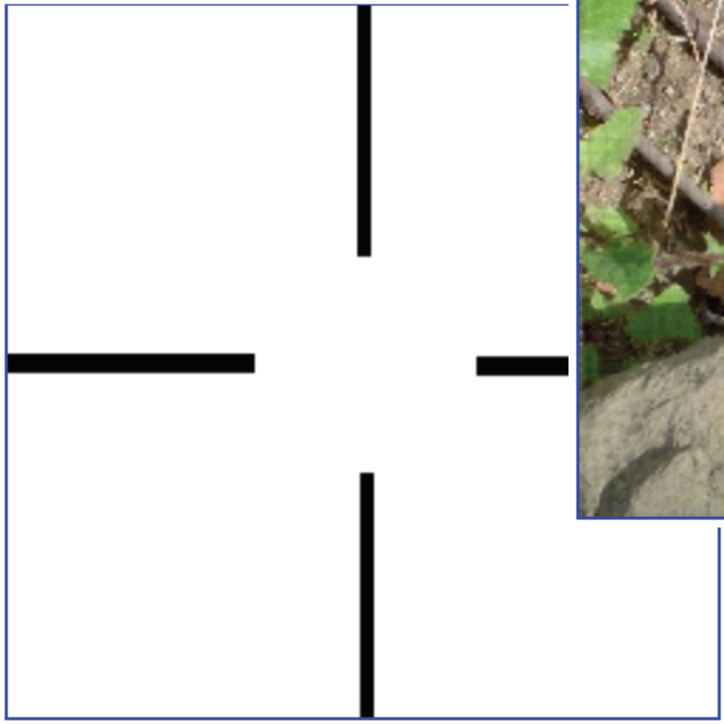
Example FFT on Images

- How about removing this 'screen' overlay from the image below? Screen is very high frequency and only in X/Y so like the mesh we will experiment with this week, can easily filter!

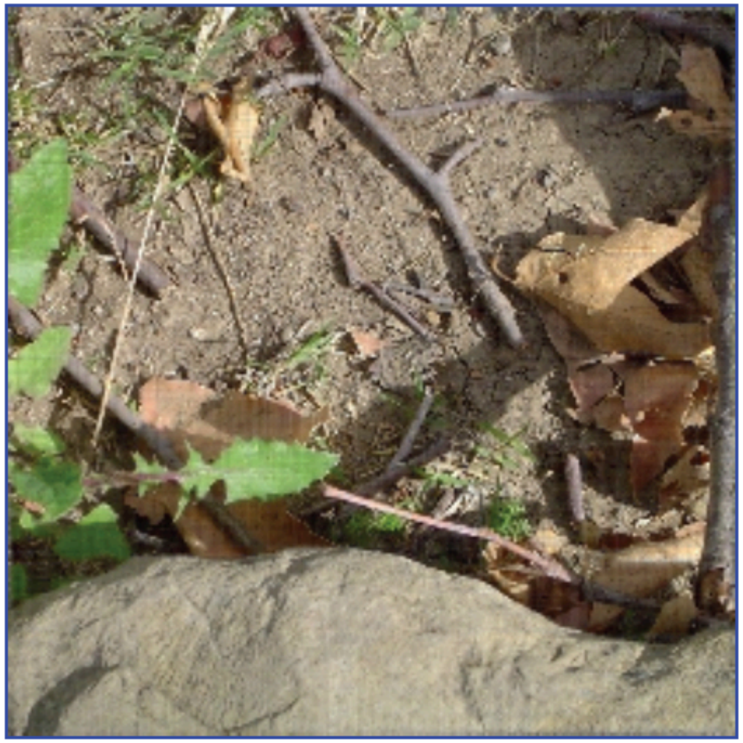
Original image.



Fourier filter.

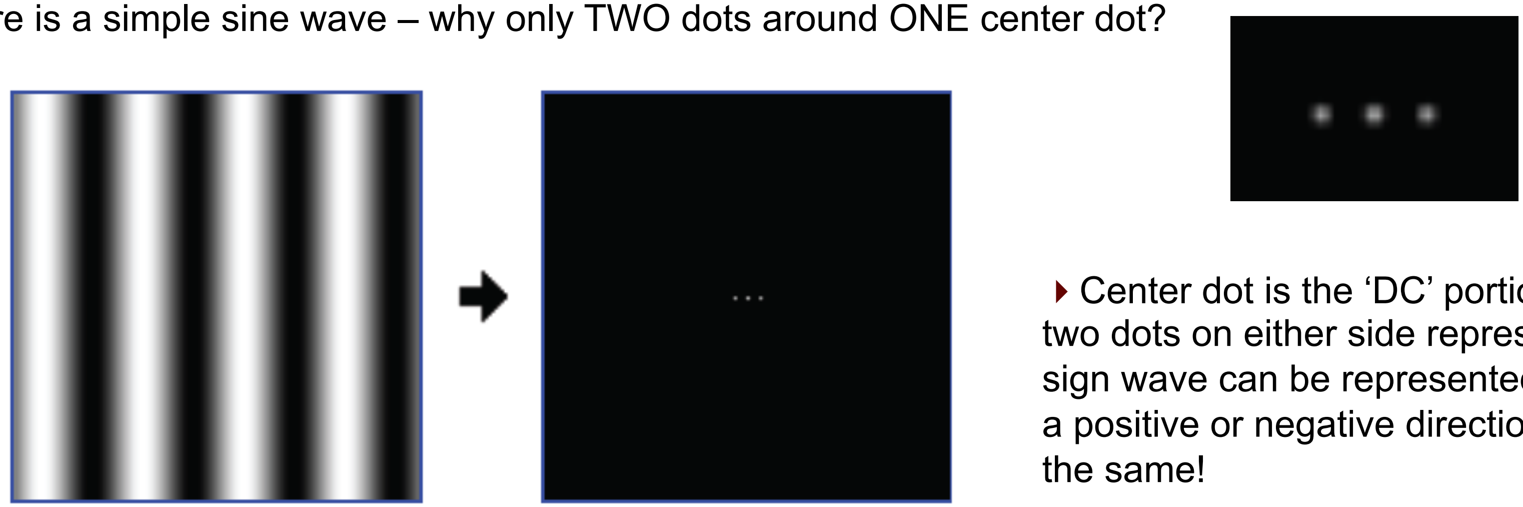


Final image!



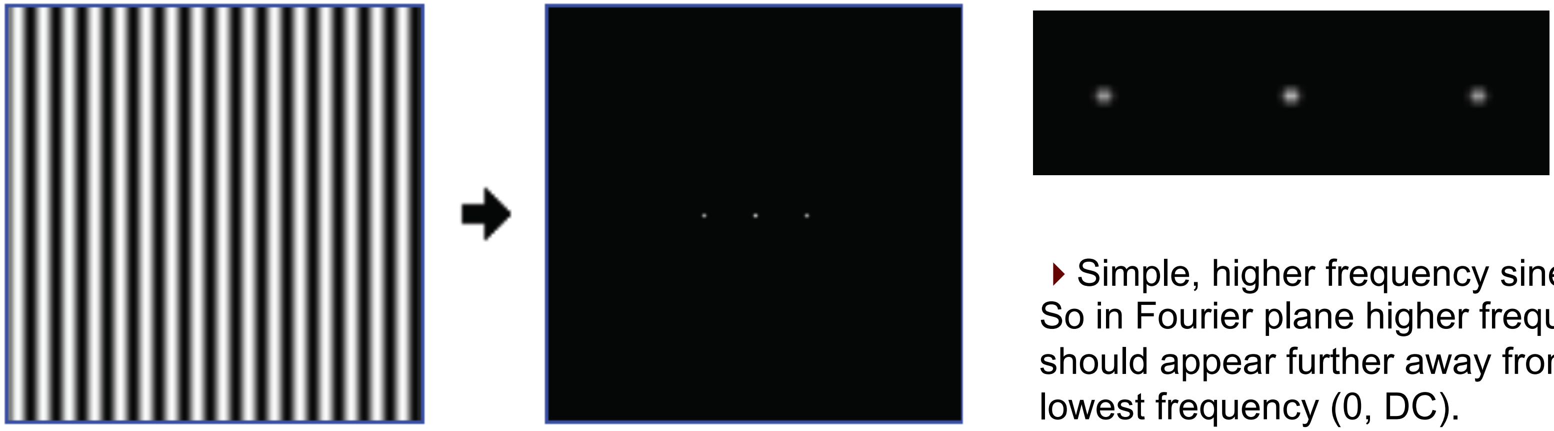
Example FFT on Images

- Here is a simple sine wave – why only TWO dots around ONE center dot?



- Center dot is the 'DC' portion, and the two dots on either side represent that a sign wave can be represented/drawn in a positive or negative direction and look the same!

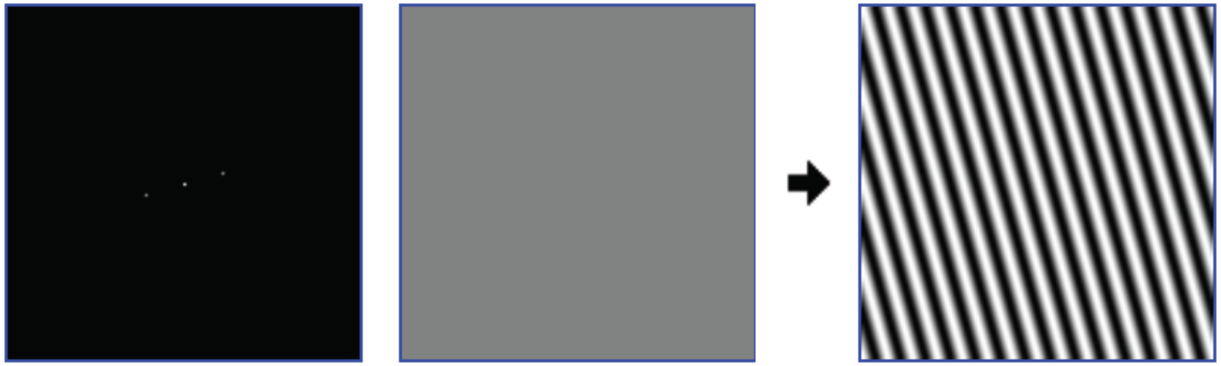
- Do it again with different sine wave… why dots further out now?

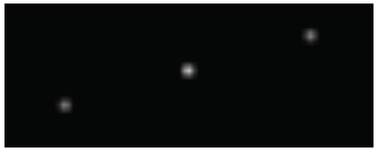


- Simple, higher frequency sine wave! So in Fourier plane higher frequencies should appear further away from the lowest frequency (0, DC).

Example FFT on Images

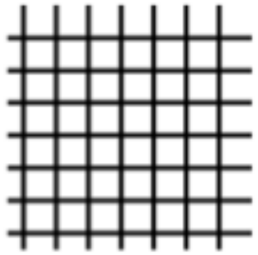
- Here you just put two dots and use to create a tilable image… wow, that is a lot of info coming out of 3 dots!
- You actually only needed one of the outer dots (as they are mirror images of each other).

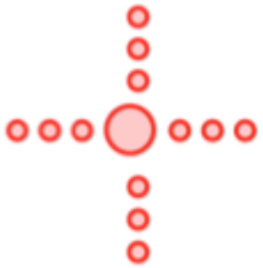




- Any opportunities in this image for compression?
- 750 x 558 pixels, 1.3 MB in .bmp and 94 kb in .jpg!
- Compression is this good or better for any image (just used this one to help you 'visualize' larger harmonics…)

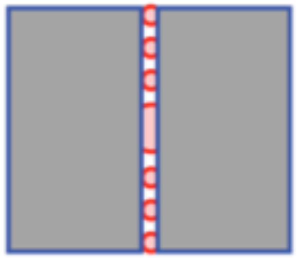
Review, Then Finished!

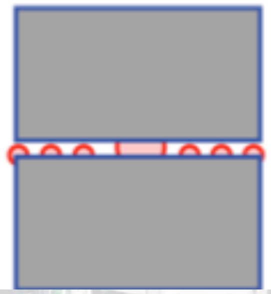




- We use a lens this week to focus light into the Fourier plane, which 3 of the following are true (one is not true).

(a) Any lens with a positive focal length will work.
(b) All the DC portion of the light collapses to the focal point because the incoming light is collimated (all parallel rays).
(c) A negative lens would work, just observe the Fourier plane at the focal point.
(d) The AC portions are simply light that is diffracted and therefore traveling in a non-collimated manner (at an angle).





- Of the two filters in the Fourier plane shown at right, which will BLOCK the vertical lines in the object (the mesh)?

(a) The vertical filter (top one).
(b) The horizontal filter (bottom one).
(c) Both.
(d) Neither.

- You want to create a system that captures an image and at the speed of light only shows the sharp edges of objects in the image (removes all 'DC' components such as uniformly colored areas, etc.). What would your Fourier filter look like?

(a) a simple opaque dot at the focal point of the lens used.
(b) a vertical or horizontal filter slit at the focal point of the lens used.
(c) a simple transmissive hole at the focal point of the lens used.
(d) All the above could work.