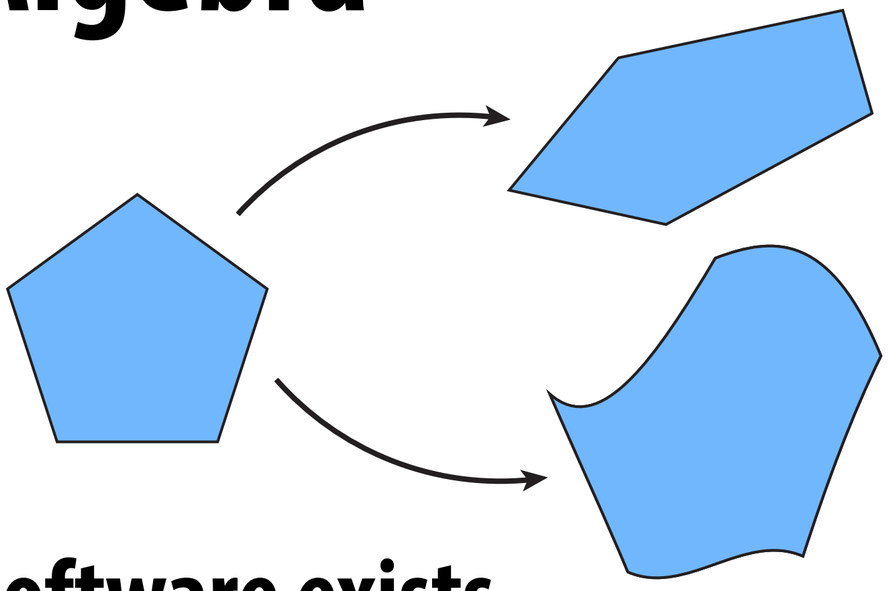
Lecture 22:

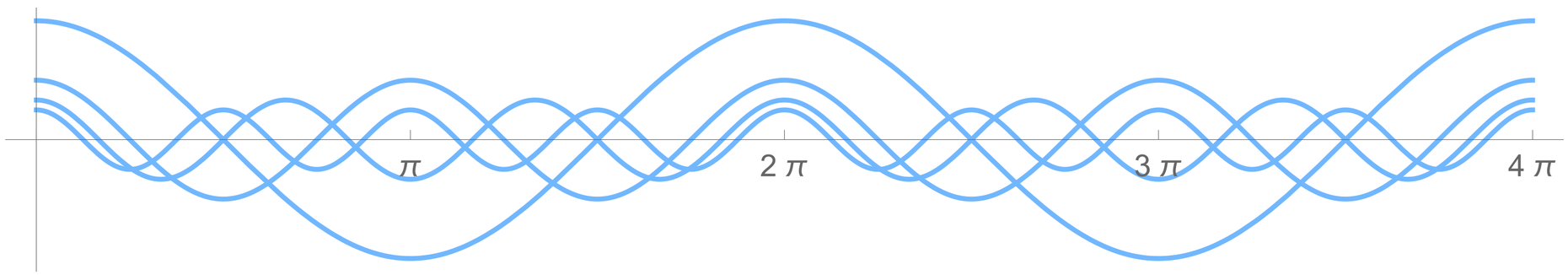
Frequency, Vibration, and Fourier

Computer Graphics
CMU 15-462/15-662, Fall 2015

Last time: Numerical Linear Algebra



- Graphics via linear systems of equations
- Why linear? Have to solve BIG problems
  - Anything else is WAY too hard (3SAT)
- Plus, lots of great (free!) linear algebra software exists
- TODAY: Fourier transform
  - Closely connected to our linear story
  - In some sense just a *change of basis*
  - But, extremely powerful in terms of computation
  - General theme in CS: change representation; things get easier



Intuition: Those knobs on your boom box

when you want to *dance!*

What are they really doing?

Intuition: Those knobs on your images

- Can do something similar with images:

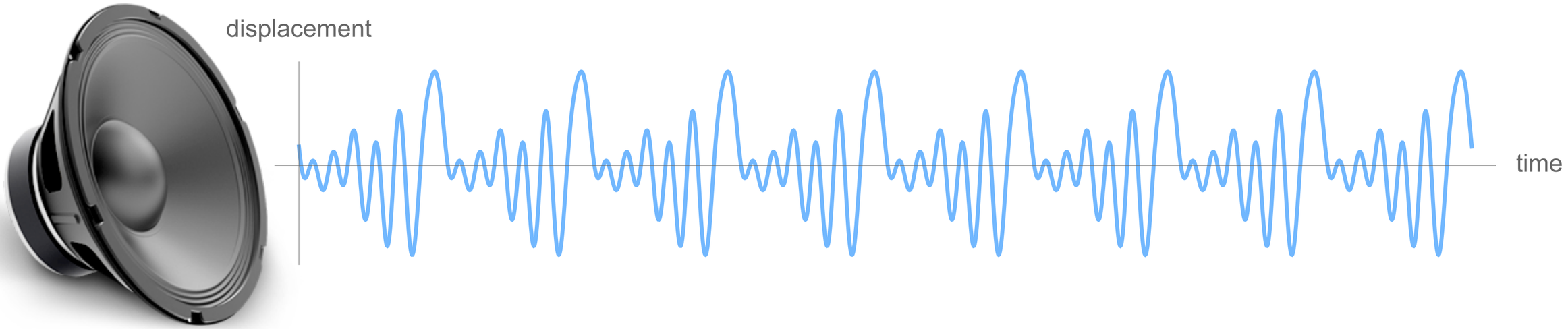
(original signal)

more "bass"

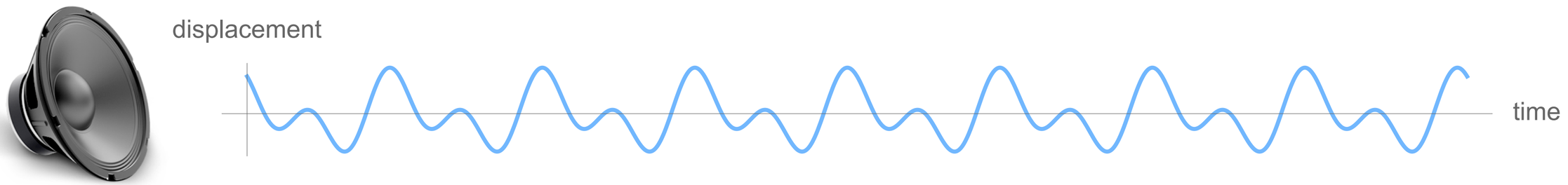
more "treble"

What's going on there?

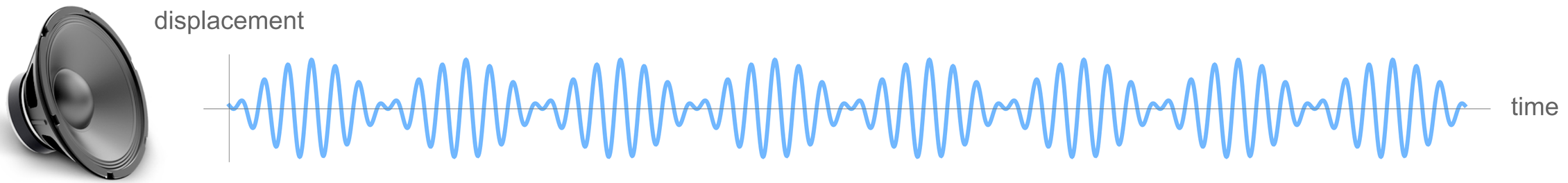
- Can think of audio signal as a function f(t): speaker cone displacement versus time:



- More "bass": kill high frequencies, keep low frequencies:



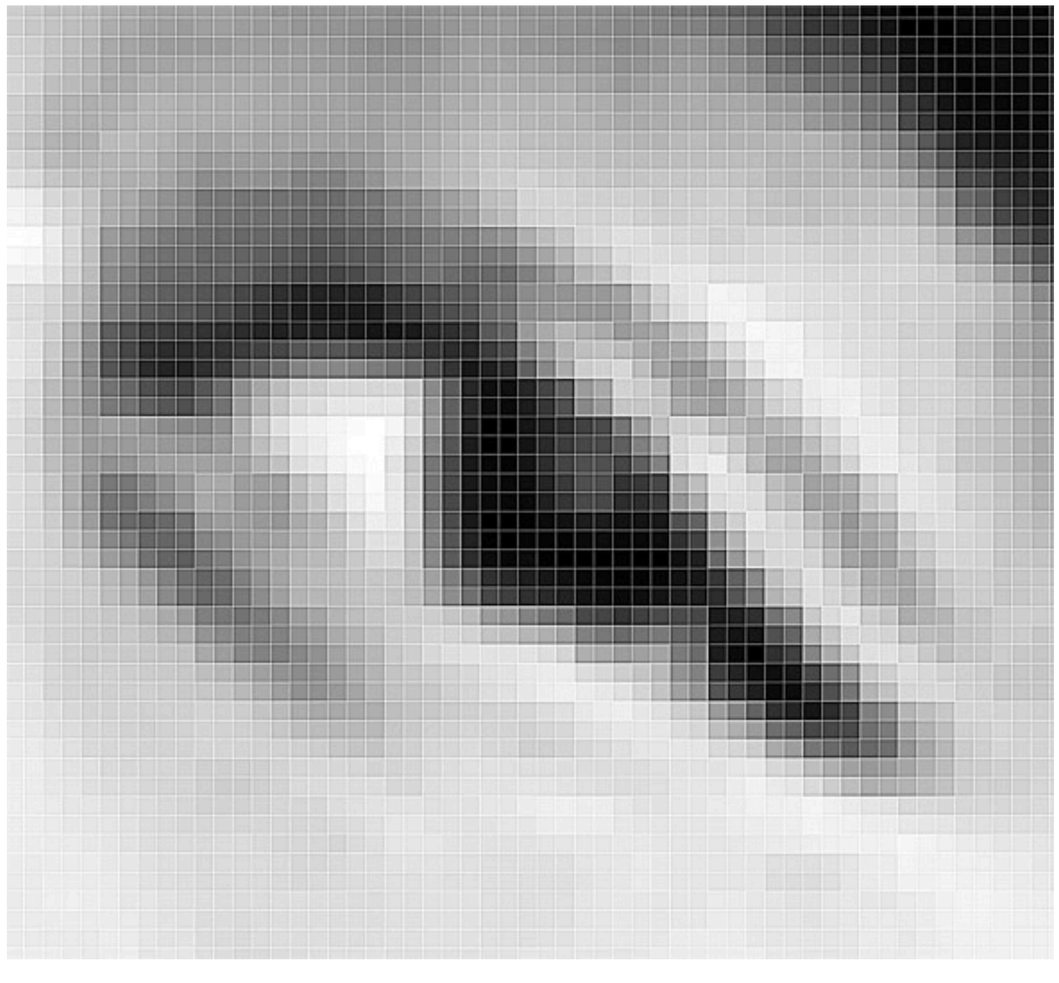
- More "treble": kill low frequencies, keep high frequencies:



How do we manipulate frequencies?

Time/Space vs. Frequency Domain

- Signal usually samples in temporal or spatial domain:
  - SOUND: amplitude as a function of time
  - IMAGE: brightness as function of space
- Not obvious how to adjust individual samples to alter *frequency*
- Change *representation*:
  - time/space => frequency
  - linear transformation!
  - (change of basis)
- Hard tasks now become easy...



Fourier Transform: Greatest Hits

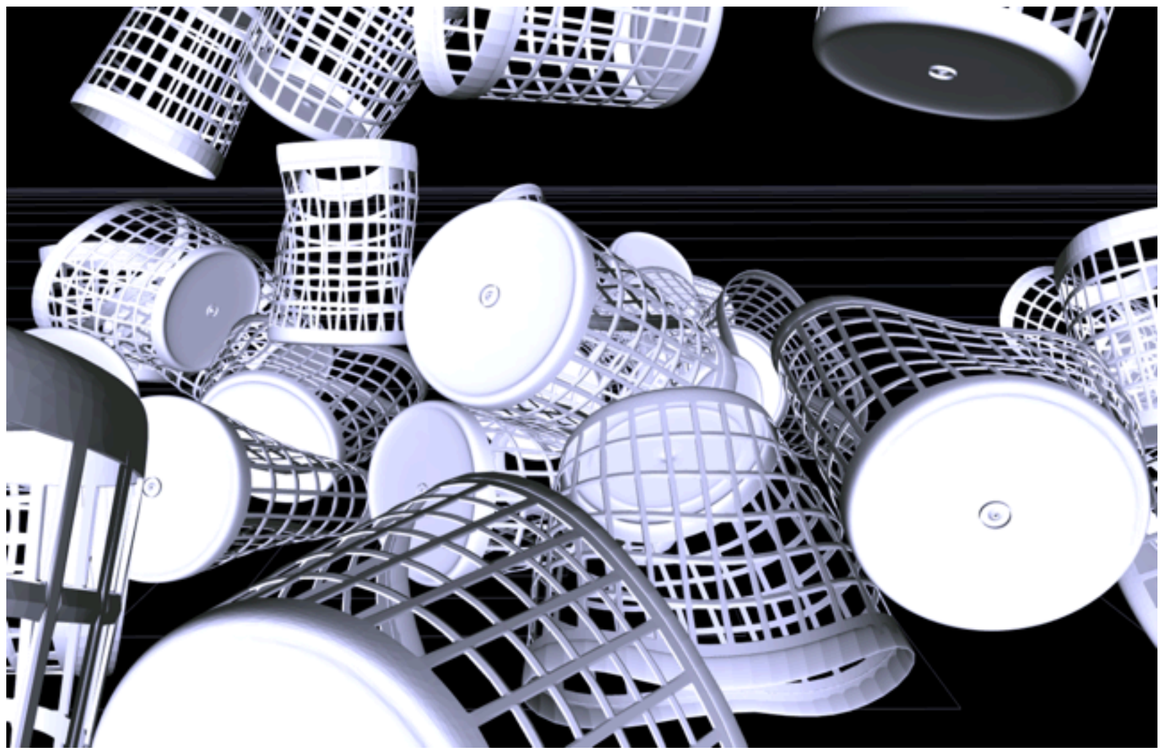
- Including revolutionary approaches to:
  - I. Filtering
  - II. Convolution
  - III. Compression
  - IV. Linear PDEs
- (...plus much, much more!)

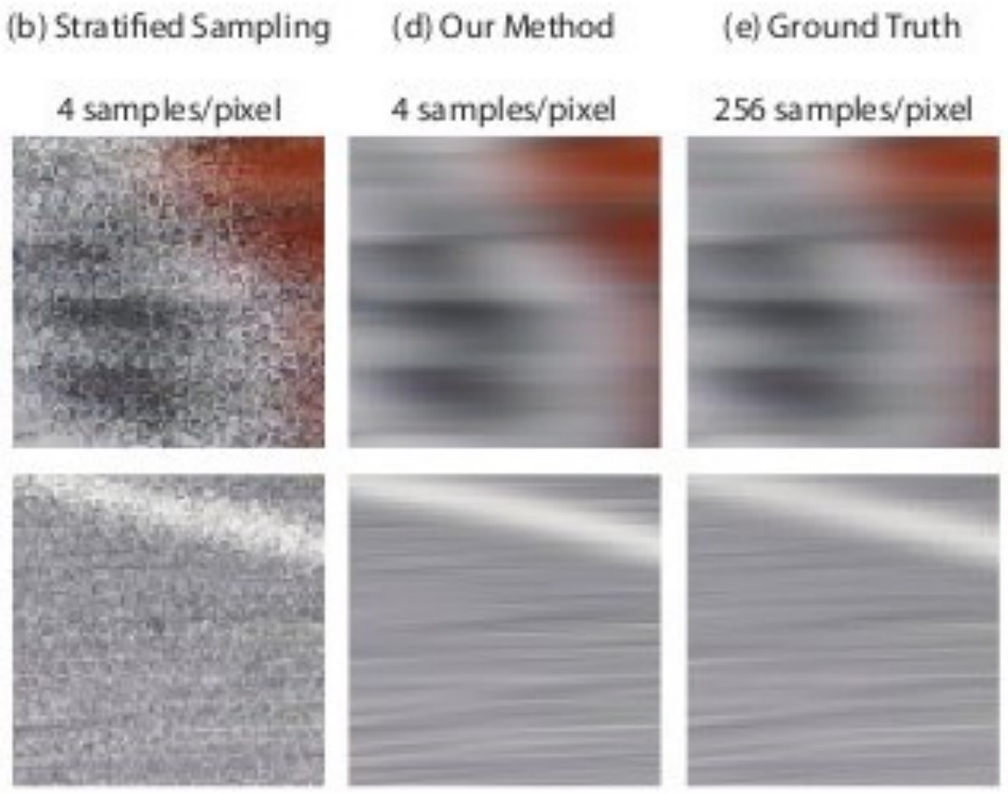
"The most important numerical algorithm of our lifetime"

—Gilbert Strang

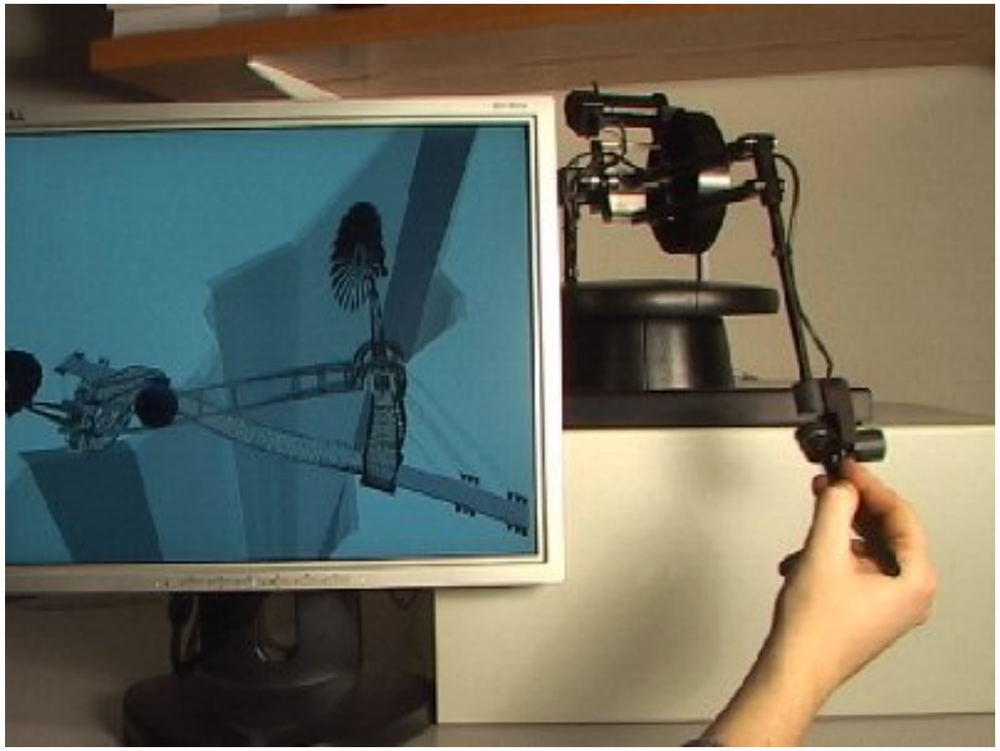
Spectral Transforms in Computer Graphics

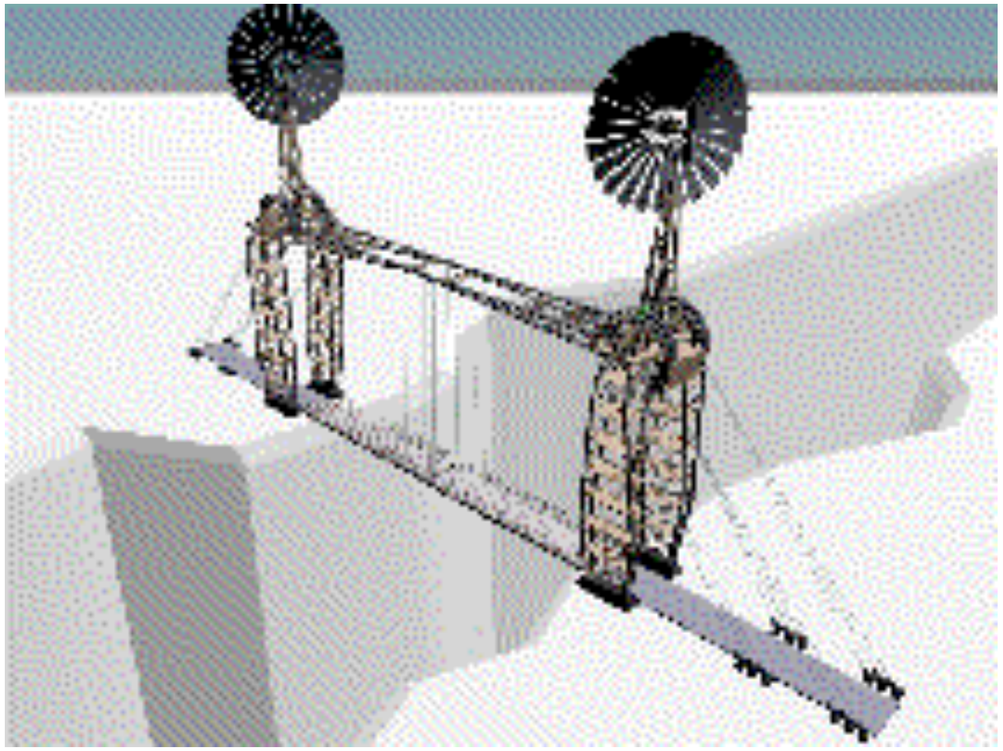
- **Powerful perspective for all sorts of *visual* problems:**
  - ANIMATION: equations of motion via fundamental modes
  - RENDERING: more intelligent sampling (see also: Shannon)
  - IMAGE PROCESSING: filter/compress images
  - GEOMETRY: filter/compress geometric features

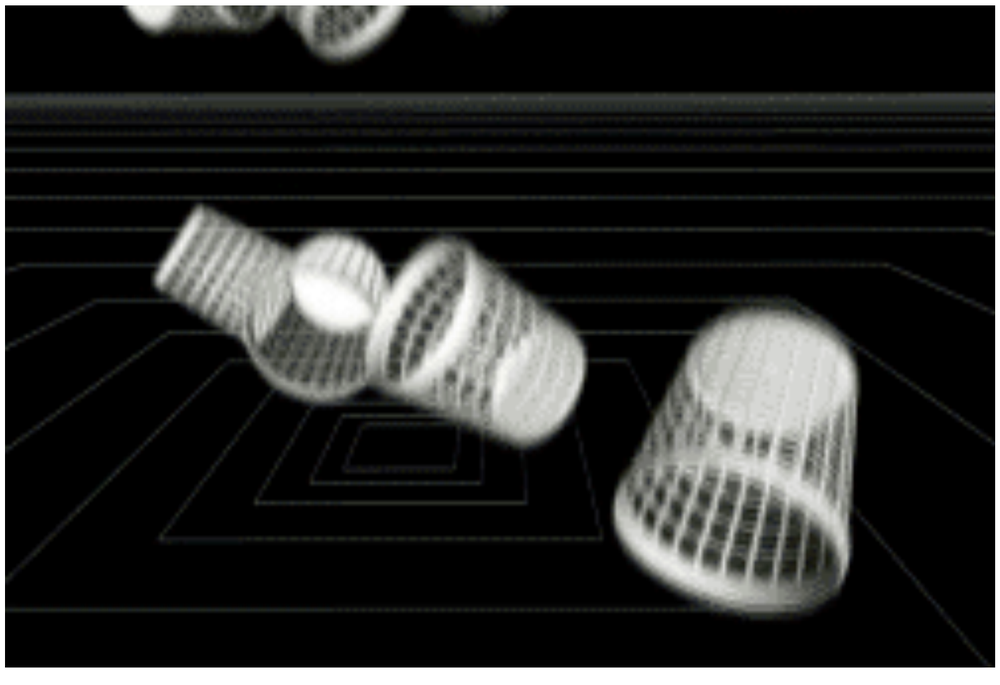


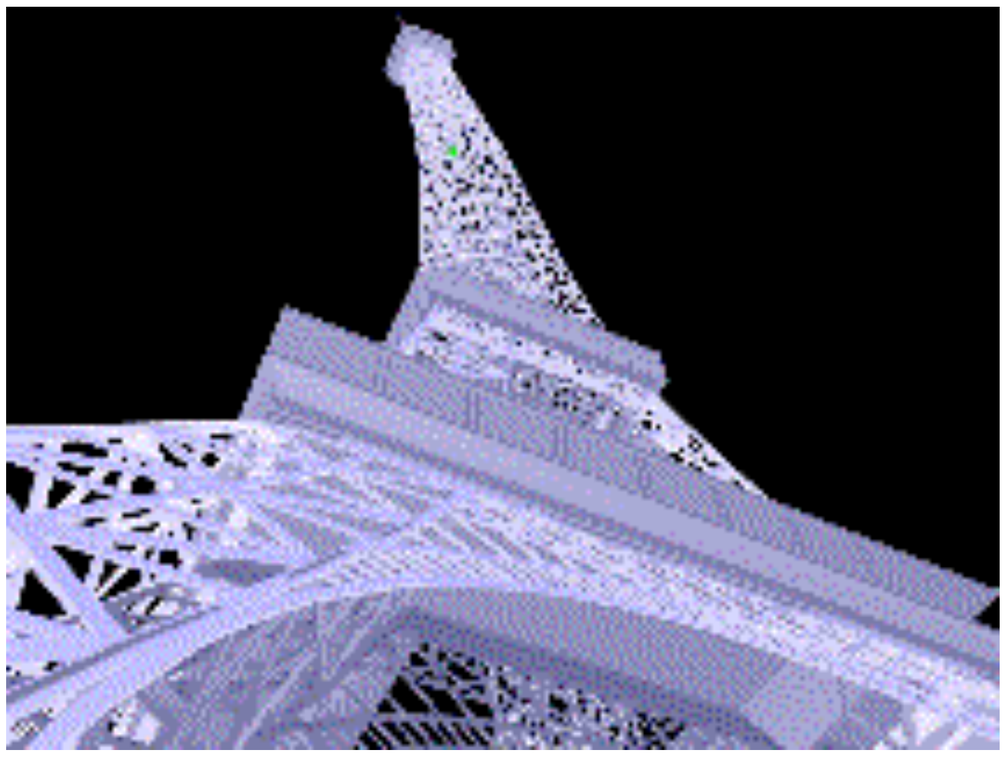


Example: Modal Simulation







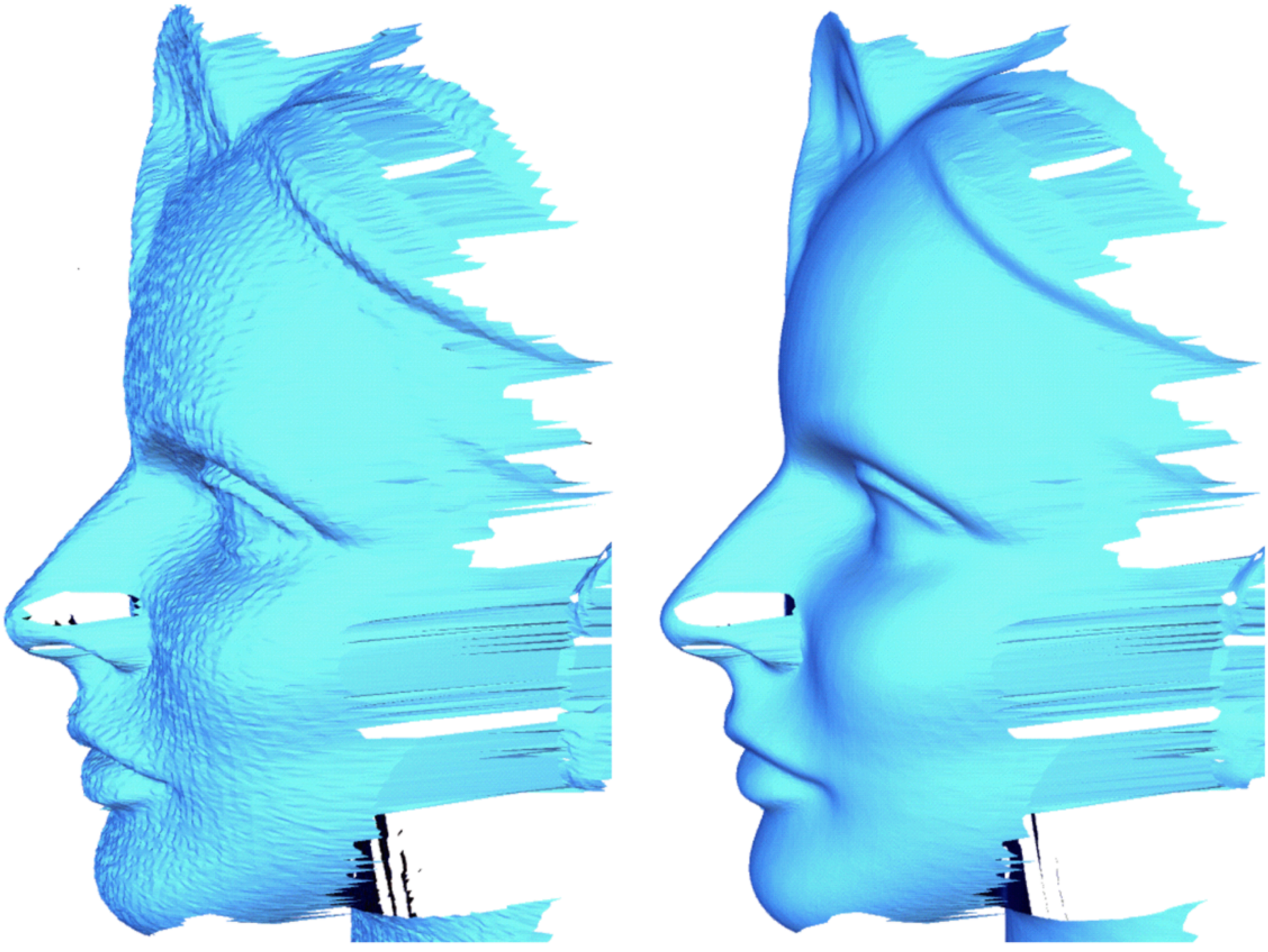


http://graphics.cs.cmu.edu/projects/stvk/

Example: Sound Synthesis

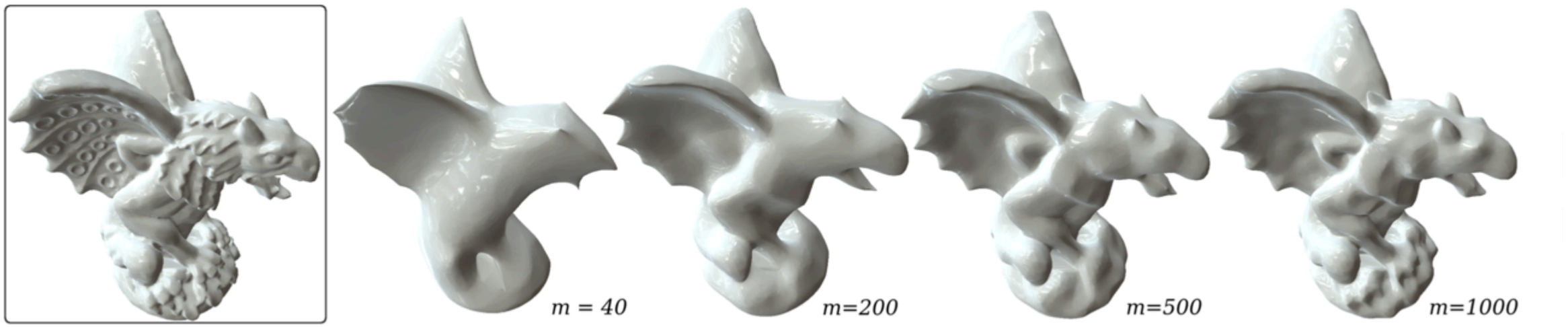
https://youtu.be/BjZ7CV6gill

Example: Mesh Filtering

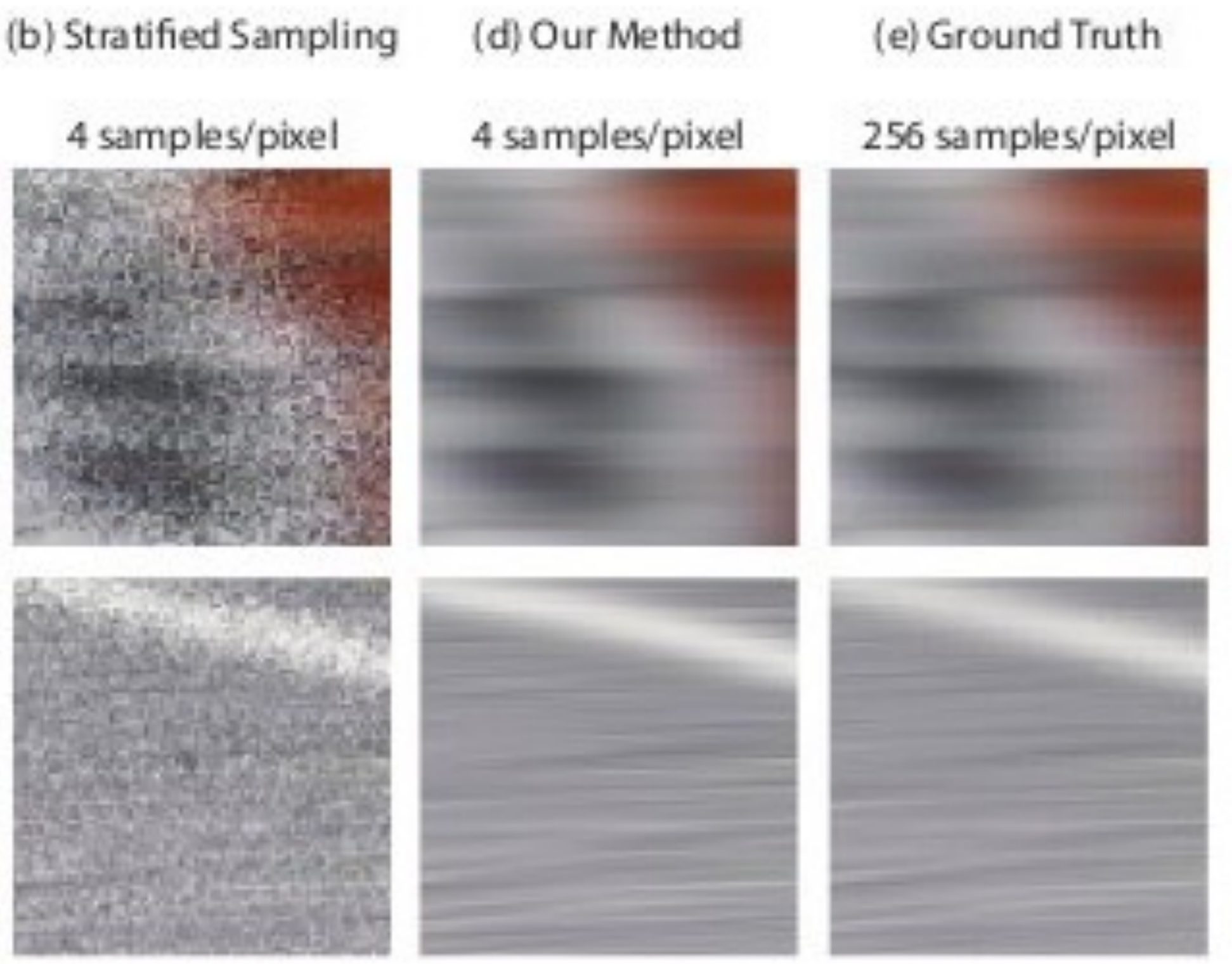


Desbrun et al, “Implicit Fairing of Irregular Meshes using Diffusion and Curvature Flow”

Example: Spectral Mesh Compression



Example: Fourier Analysis in Rendering

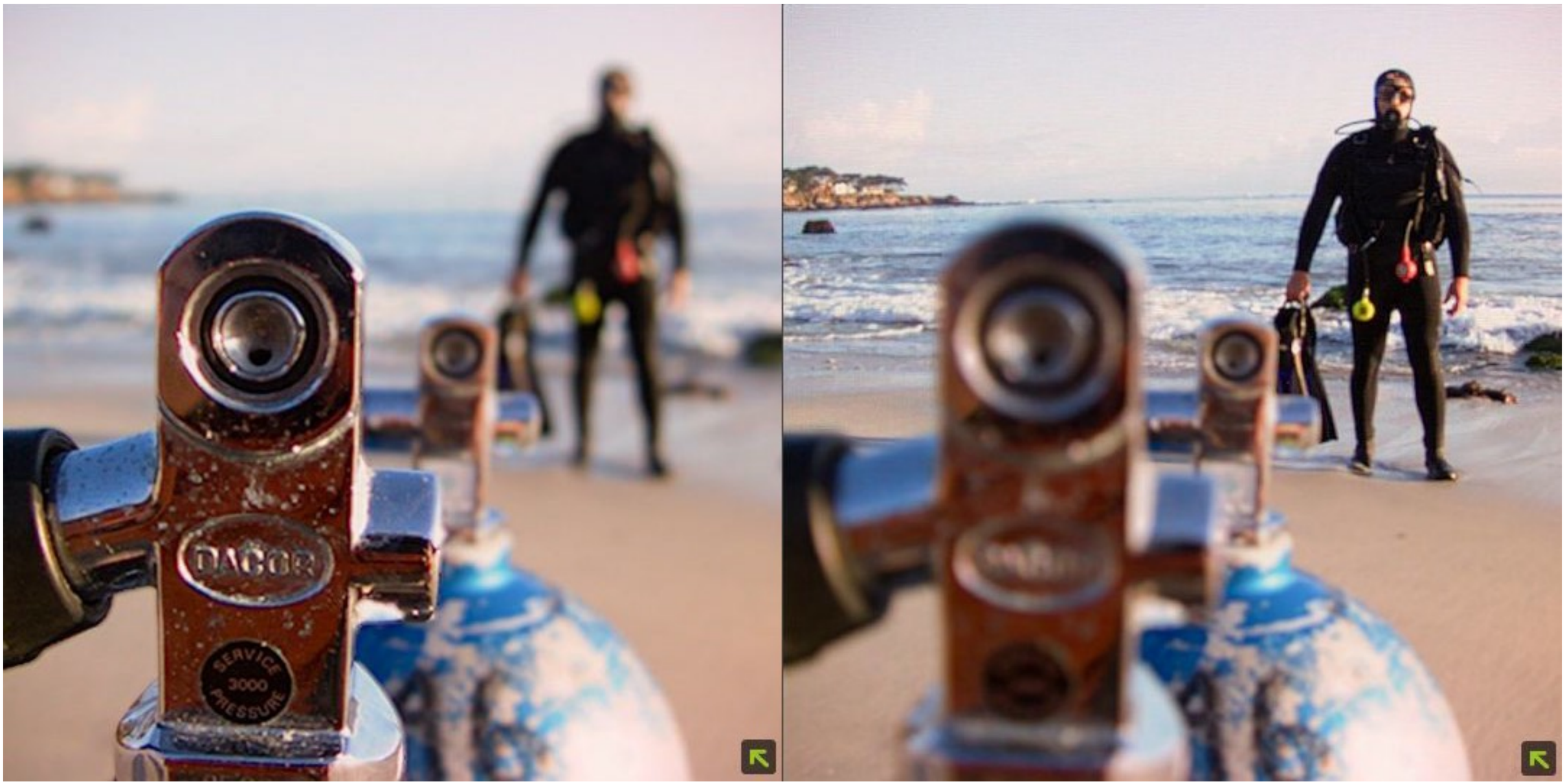


Egan et al, "Frequency Analysis and Sheared Reconstruction for Rendering Motion Blur"

Example: Motion Amplification

http://people.csail.mit.edu/mrub/vidmag/

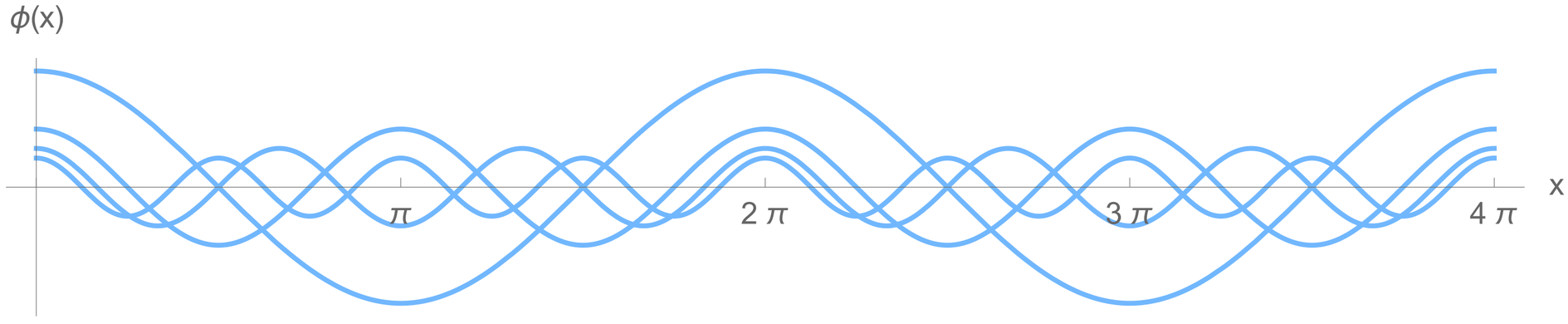
Example: Light Field Photography



http://www.lytro.com/

What are “frequencies”?

- How do we talk about frequencies mathematically?
- Natural idea in 1D: sinusoidal functions*



$$\phi(t) = \cos(\omega t + \varphi)$$

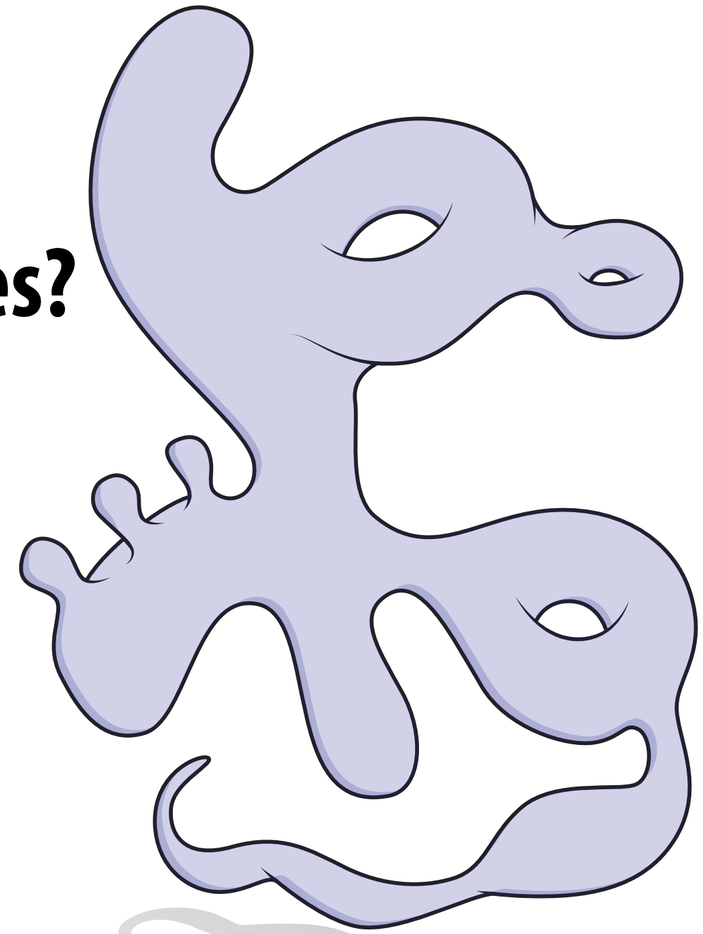
frequency
“how often does it repeat?”

phase
“where does it begin?”

- Real line: can have any real frequency
- Periodic interval [0,2π): frequency must be integer

*drawn at different amplitudes only for purpose of visualization!

Sinusoids, more generally



- What about other domains, like curved surfaces?
- Does it make sense to talk about frequencies?
- Well, notice what happens if we take the 2nd derivative of a sinusoid:

$$\phi(t) = \cos(\omega t + \varphi)$$

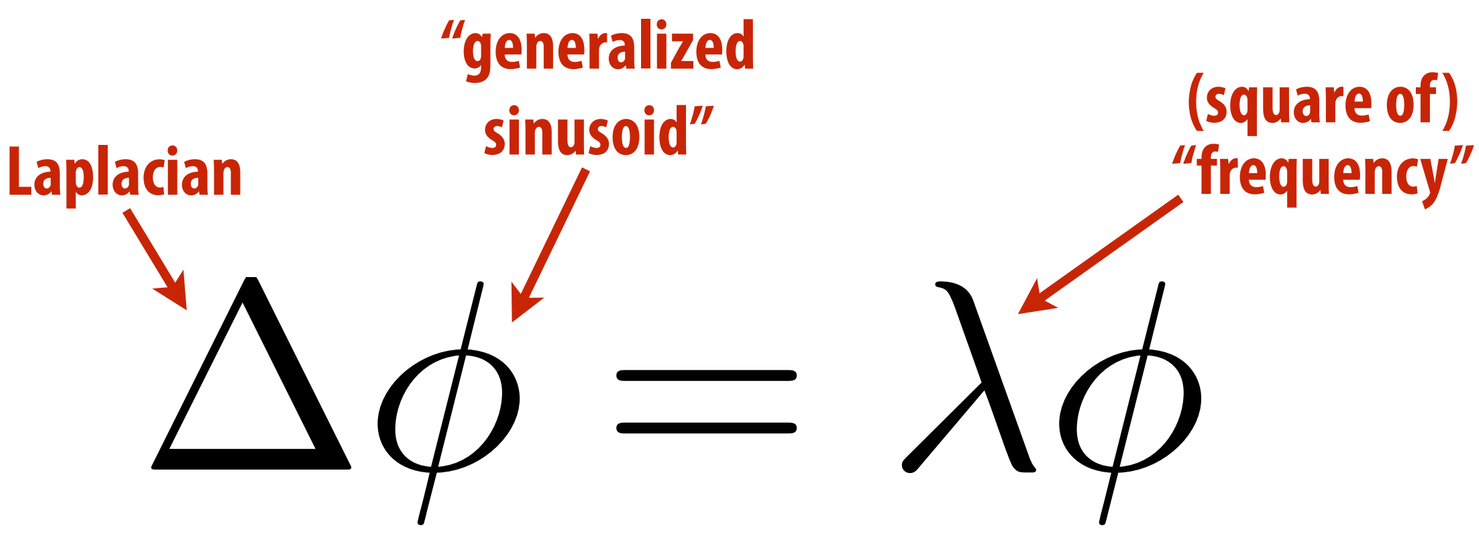
$$\Rightarrow \frac{\partial^2}{\partial t^2}\phi = -\omega^2 \cos(\omega t + \varphi)$$

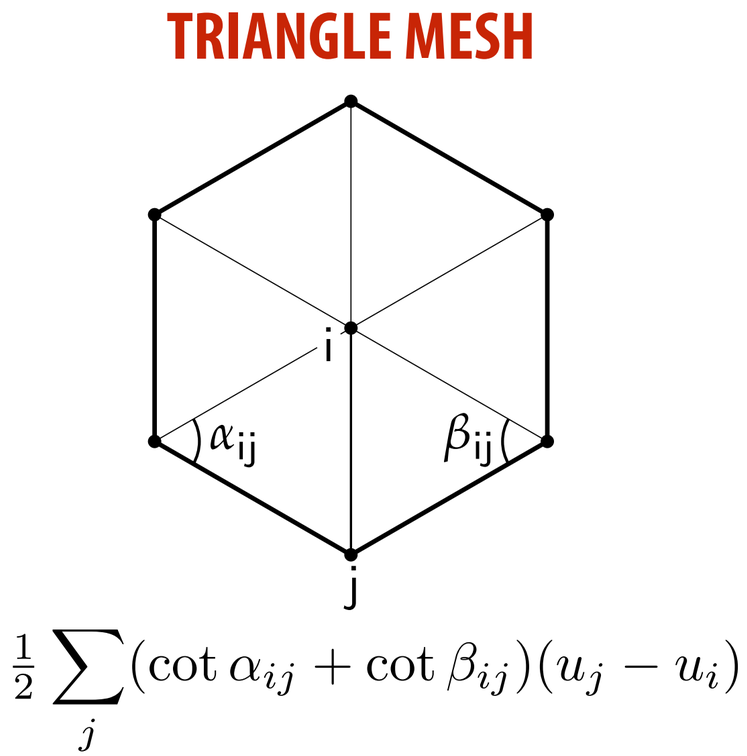
- Hmm... sinusoids appear to be *eigenfunctions* of 2nd derivative!
- I.e., 2nd derivative is just the same function, up to a rescaling
- And recall that in 1D, 2nd derivative is the *Laplace operator:*

$$\frac{\partial^2}{\partial t^2} \iff \Delta$$

Laplacian Eigenfunctions as Spectral Bases

- Turning that around: we have Laplace on any domain
- So, we can *define* "sinusoids" as Laplacian eigenfunctions:





- In traditional signal processing, this is an "advanced" topic
- In computer graphics, it is a "necessary" topic!
- Why? Because computer graphics has to deal with domains that are not flat/square! (Curves, surfaces, volumes, ...)

Laplacian Eigenfunctions, Visualized

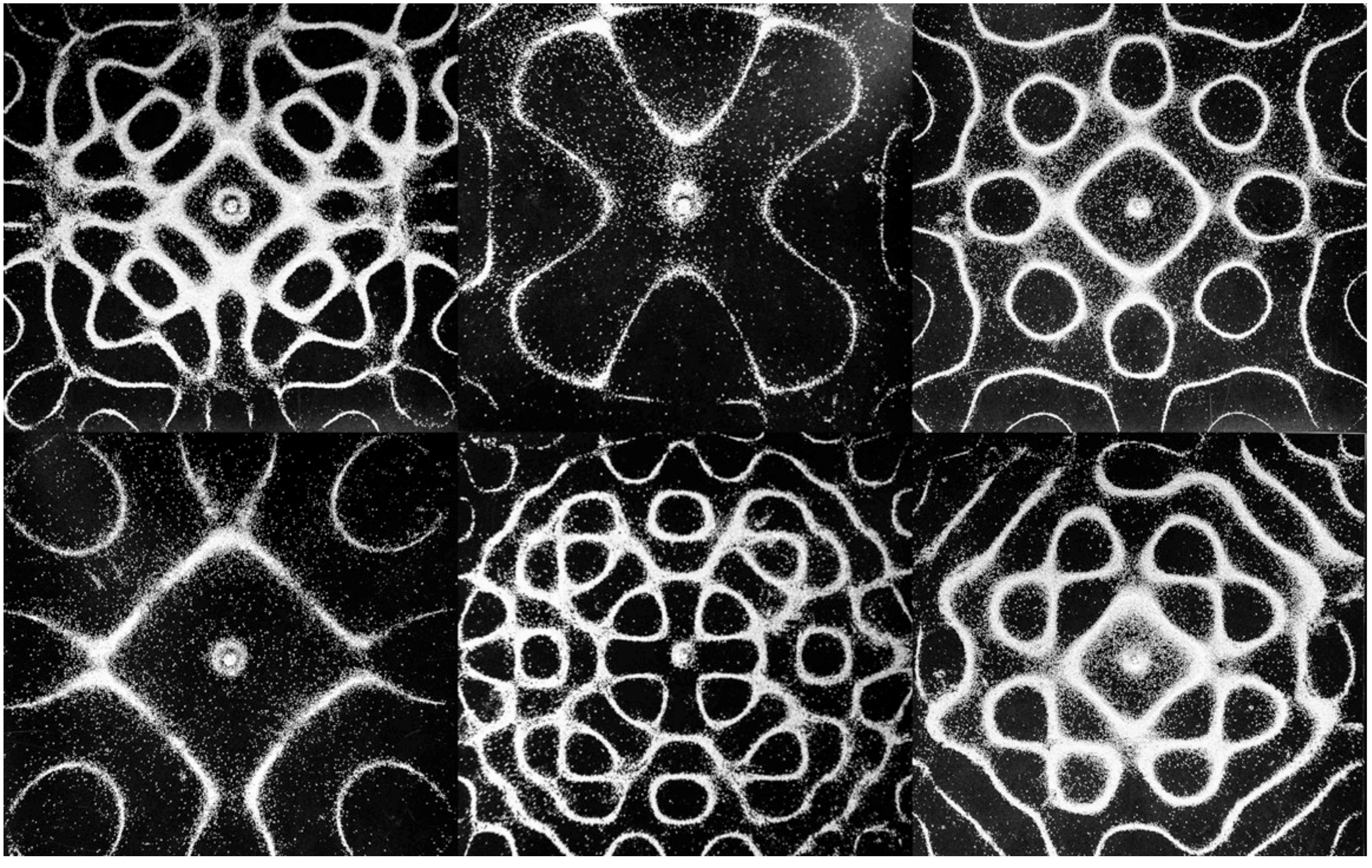
(*Some* solutions to $\Delta\phi = \lambda\phi$):

lower frequency

higher frequency

Laplacian Eigenfunctions, IRL

- In fact, these patterns show up all over the natural world:



"Chladni plates"

Laplacian Eigenfunctions, IRL

- Used to analyze geometry/shape long before computers:

Ok, but why are these functions *useful?*

Claim: Any* function can be expressed as a sum of sinusoids.

*"sufficiently nice"

Review: Inner Product

- To take this idea any further, we will need an *inner product*
- Last time, defined a *vector space* (What was that…?)
- On top of any vector space, we can define an *inner product*
- Intuition: measures how much two vectors "line up"
- More formally, eats two vectors and spits out a number:

$$\langle \cdot, \cdot \rangle : V \times V \to \mathbb{R}; (u, v) \mapsto \langle u, v \rangle$$

- …and does so in a "natural" way:

$\langle u, v \rangle = \langle v, u \rangle$ amount things line up doesn't depend on order!

$\langle au, v \rangle = a\langle u, v \rangle$ scaling a vector just scales the inner product

$\langle u + v, w \rangle = \langle u, w \rangle + \langle v, w \rangle$ …altogether, linear in each vector ("bilinear")

$\langle x, x \rangle \geq 0, \quad 0 \iff x = 0$ also, nonzero vectors have nonzero length!

Review Inner Product on R^n

- **Suppose you have two vectors in R^n:**

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \qquad y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

Euclid of Alexandria

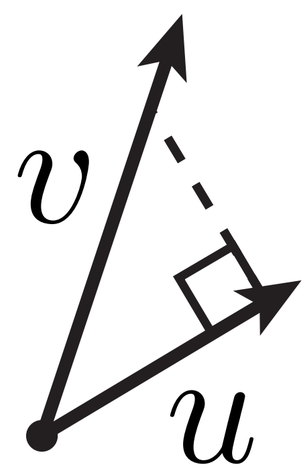
- **Q: How do you (usually) take their inner product?**

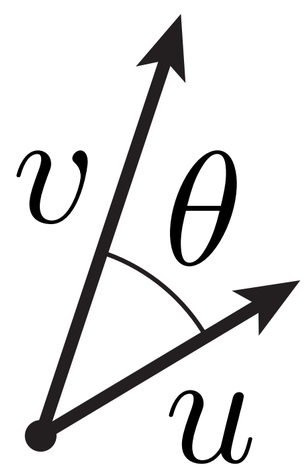
$$\langle x, y \rangle = x_1 y_1 + \cdots + x_n y_n = \sum_{i=1}^{n} x_i y_i$$

- **Often called the "Euclidean" inner product, because there was a really old Greek dude called Euclid who thought a lot about how measurement works in flat spaces like the plane.**

Inner Product, Visualized

- Don't get too caught up in the notation: like everything in linear algebra, inner product has concrete geometric meaning*:





$$\langle v, u/|u| \rangle \qquad \langle u, v \rangle = |u||v| \cos\theta$$

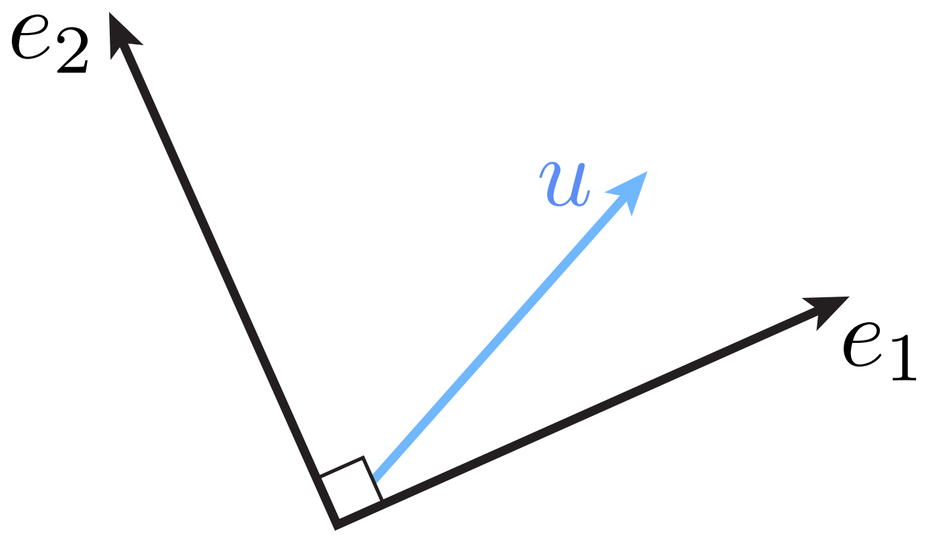
- This first picture (projection of one vector onto another) will be *very important* for understanding the Fourier transform.

*Euclid himself did not have modern algebra to work with, but he *did* understand the geometry!

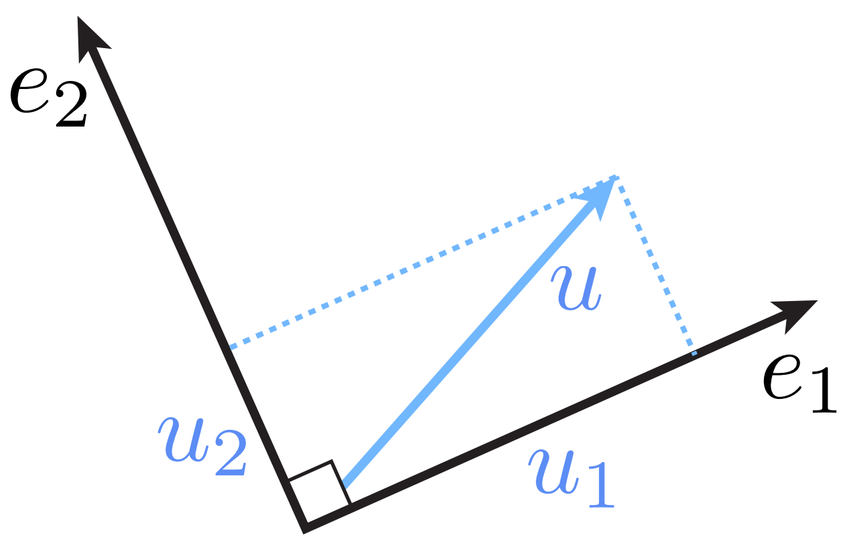
Projecting onto a Basis

a.k.a., a *basis!*

- Suppose you have a vector u, and orthonormal vectors e_1, e_2:



- Q: What are the coordinates of u in this basis?
- A: Just project!



$$u_1 = \langle u, e_1 \rangle$$
$$u_2 = \langle u, e_2 \rangle$$

Fourier transform in a nutshell:

Project signal onto sinusoids.

Inner Product on Functions

- Ok, but wait… what does that mean for functions? What's a projection? And what's an inner product?
- Recall intuition: how much do two vectors (functions) "line up?"
- For vectors, we had a sum:

$$\langle x, y \rangle = \sum_{i=1}^{n} x_i y_i$$ *"Euclidean inner product"*

- Q: What about for functions?

$$\langle\langle f, g \rangle\rangle := \int_\Omega f(p) g(p) \; dp$$ *"L² inner product"*

- Careful though—integral must exist! (*"square integrable"*)

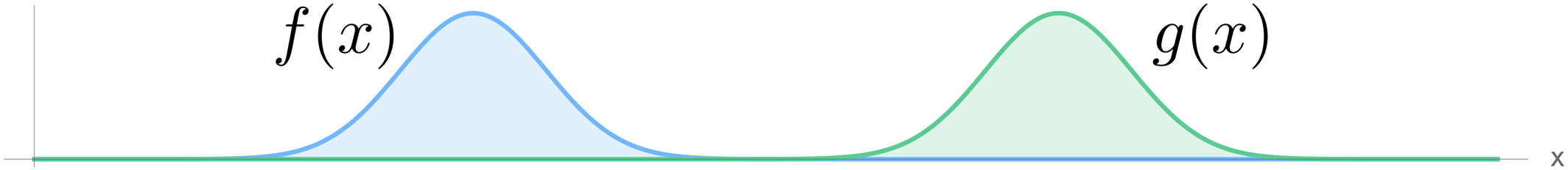
$$\langle\langle f, f \rangle\rangle = \int_\Omega f(p)^2 \; dp$$

Can only use L² inner product if this integral (squared "L² norm") is *finite*.

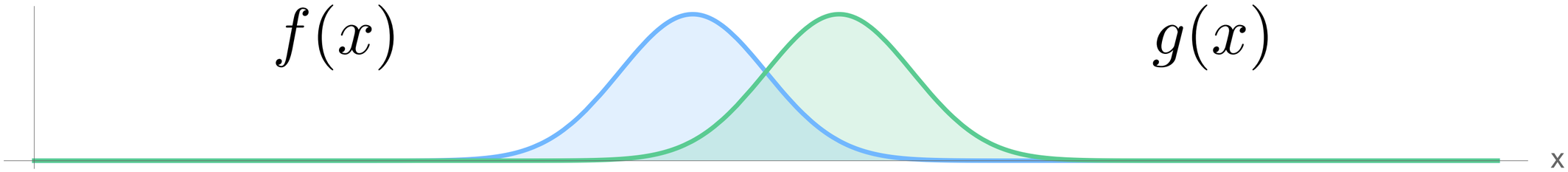
L^2 Inner Product

- **Does the L2 inner product agree with our intuition about the inner product? (Does it measure how well things "line up?")**
- **Yes! Some examples:**

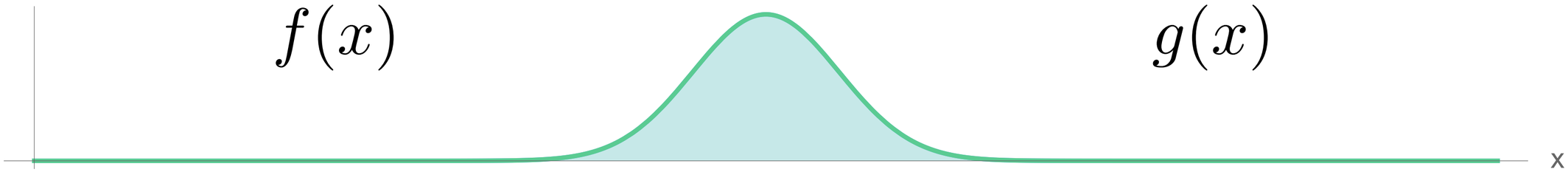
$$\langle\langle f, g \rangle\rangle := \int_\Omega f(x) g(x) \; dx$$



Almost no overlap—inner product is almost zero.



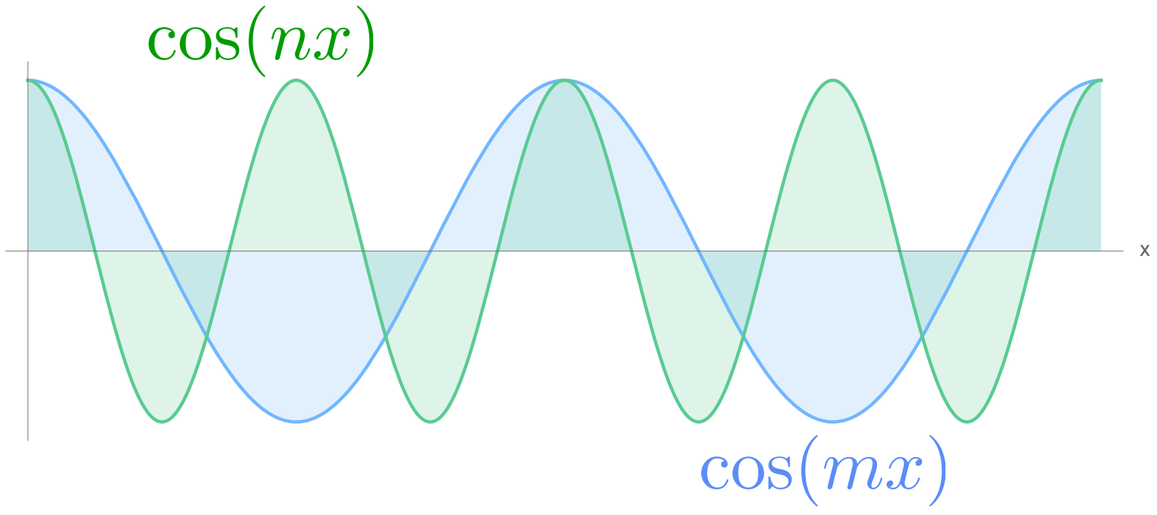
Some overlap—inner product gets bigger.

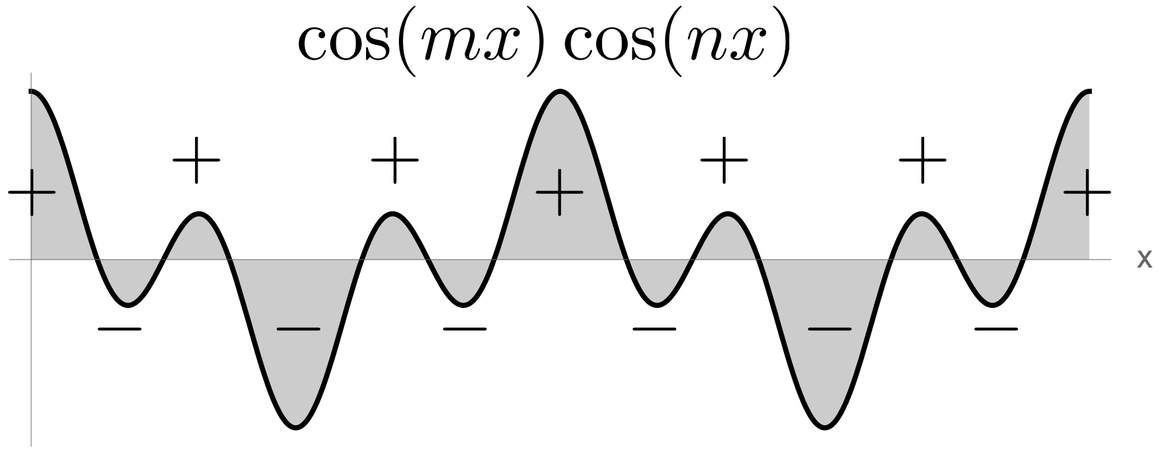


Total overlap—inner product yields norm (squared).

L^2 Inner Product of Sinusoids

- Interesting thing happens if we do the same thing w/ sinusoids:



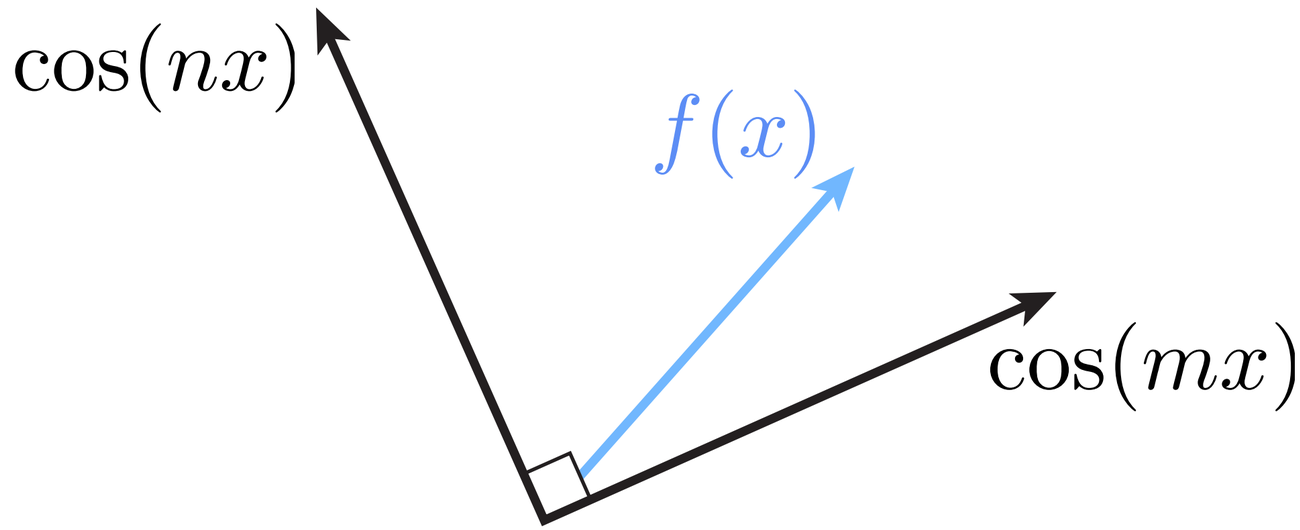


- What's the L^2 inner product in this example?
  - two big positive bumps, four small positive bumps
  - two big negative bumps, four small negative bumps
  - integral is "area under the curve"; must be zero!
- In general*:

$$\langle\langle \cos(mx), \cos(nx) \rangle\rangle = \begin{cases} 0, & m \neq n \\ \pi, & m = n \end{cases}$$

*on domain [0,2π); similar story with sin(mx)sin(nx) & cos()sin()...

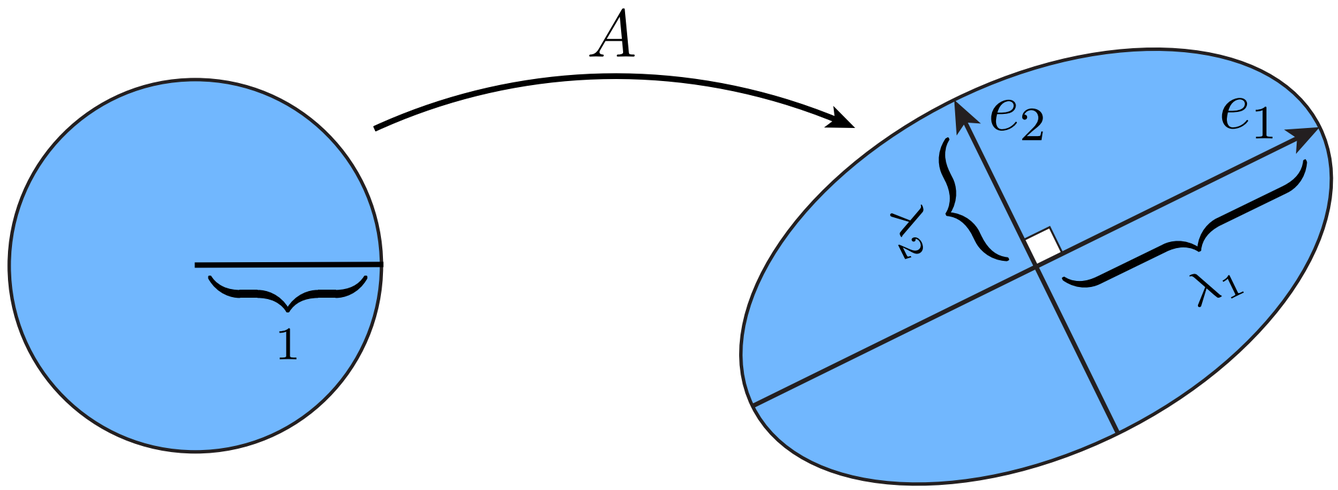
In other words, sinusoids are *orthogonal.*



(Smells like a basis.)

Spectral Theorem, Revisited

- If it looks like a basis, and smells like a basis…
- FACT: Eigenfunctions of the Laplace operator form a basis for square integrable functions.
- Actually, just a "souped-up" version of what we said about symmetric matrices last time:



$$Ae_i = \lambda_i e_i$$

- All we did was replace the matrix A with the "2nd derivative" Δ
- *Both* are maps from vectors to vectors (Δf is a function—or *vector!*)
- Of course, when we discretize, Δ becomes a matrix again!

(What goes around comes around…)

Fourier Transform—*finally*

- **So how do we do a Fourier transform of a function f?**
- **Just project it onto all the Laplacian eigenfunctions. E.g., in 1D:**

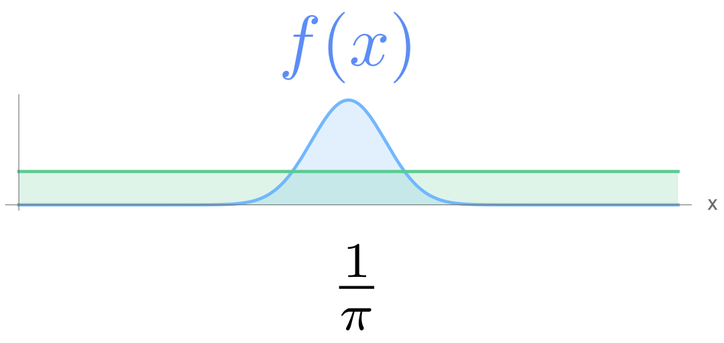
$$a_i := \langle\langle f, \cos(nx) \rangle\rangle \quad (n \in \mathbb{N})$$
$$b_i := \langle\langle f, \sin(nx) \rangle\rangle$$

1D Fourier basis

$$(a_0, b_0, a_1, b_1, \ldots)$$

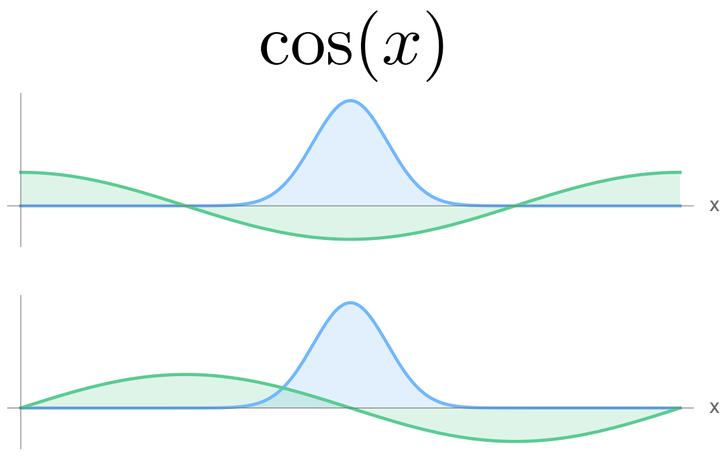
"Fourier coefficients"

- **Visually:**



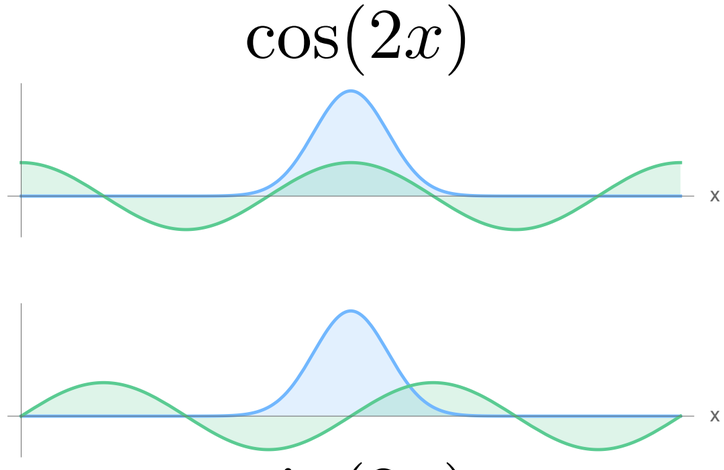
$a_0 = b_0 = 0.886227$

$a_1 = -0.832533$



$b_1 = 0$

$a_2 = 0.690194$



$b_2 = 0$

(Notice: some coefficients are small or zero… leads to *compression!*)

What if I want my original signal back?

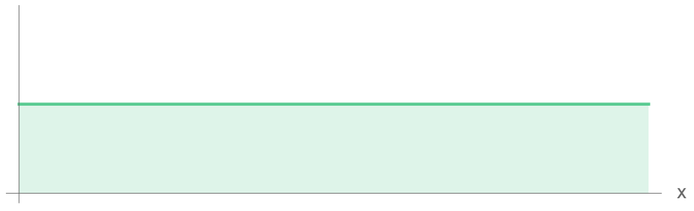
Inverse Fourier Transform

- **Given Fourier coefficients, can reconstruct by just taking weighted sum of Fourier bases:**

$$f(x) = a_0 + \frac{1}{\pi}\sum_{k=1}^{\infty} a_k \cos(kx) + b_k \sin(kx)$$

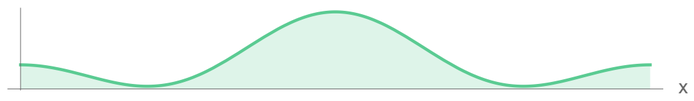
- **May take many terms before we see a good reconstruction!**

- **E.g., our example from two slides back:**

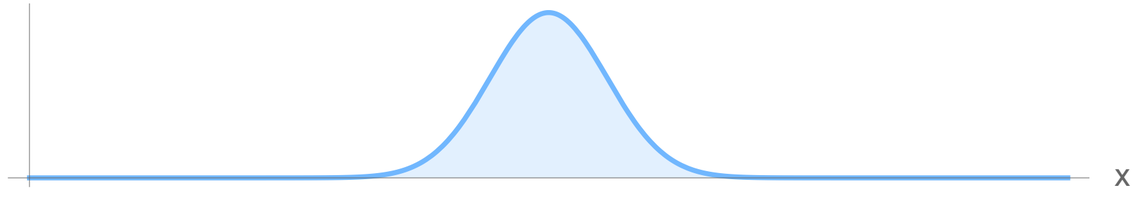


one term

two terms



three terms



(original function)

Fourier Transform in Practice

- In practice…
  - Typically work with *sampled* rather than *continuous* function
  - Conceptually, no different: project onto eigenbasis
  - Can use basis of any discrete Laplace operator
- BUT…
  - Computation is EXTREMELY expensive in general
  - E.g., unstructured triangle mesh: $O(n^3)$ or so
- Two options:
  - Approximate w/ a few Fourier bases (like last slide) or
  - Use *fast Fourier transform* (FFT)

Fast Fourier Transform

- Revolutionary algorithm not only for signal processing, but many problems in computation
- Don't have time to do it justic in this lecture!
- But, a few key points:
  - Cost is O(n log n) in 1D
  - Cost is O(m log m) in 2D, where $m = n^2$
  - *Extremely* well-developed software available
  - E.g., FFTW ("Fastest Fourier Transform in the West")
  - *Can also be done on the sphere!* ("spherical harmonics")

"The most important numerical algorithm of our lifetime"

—Gilbert Strang

Slow Fourier Transform

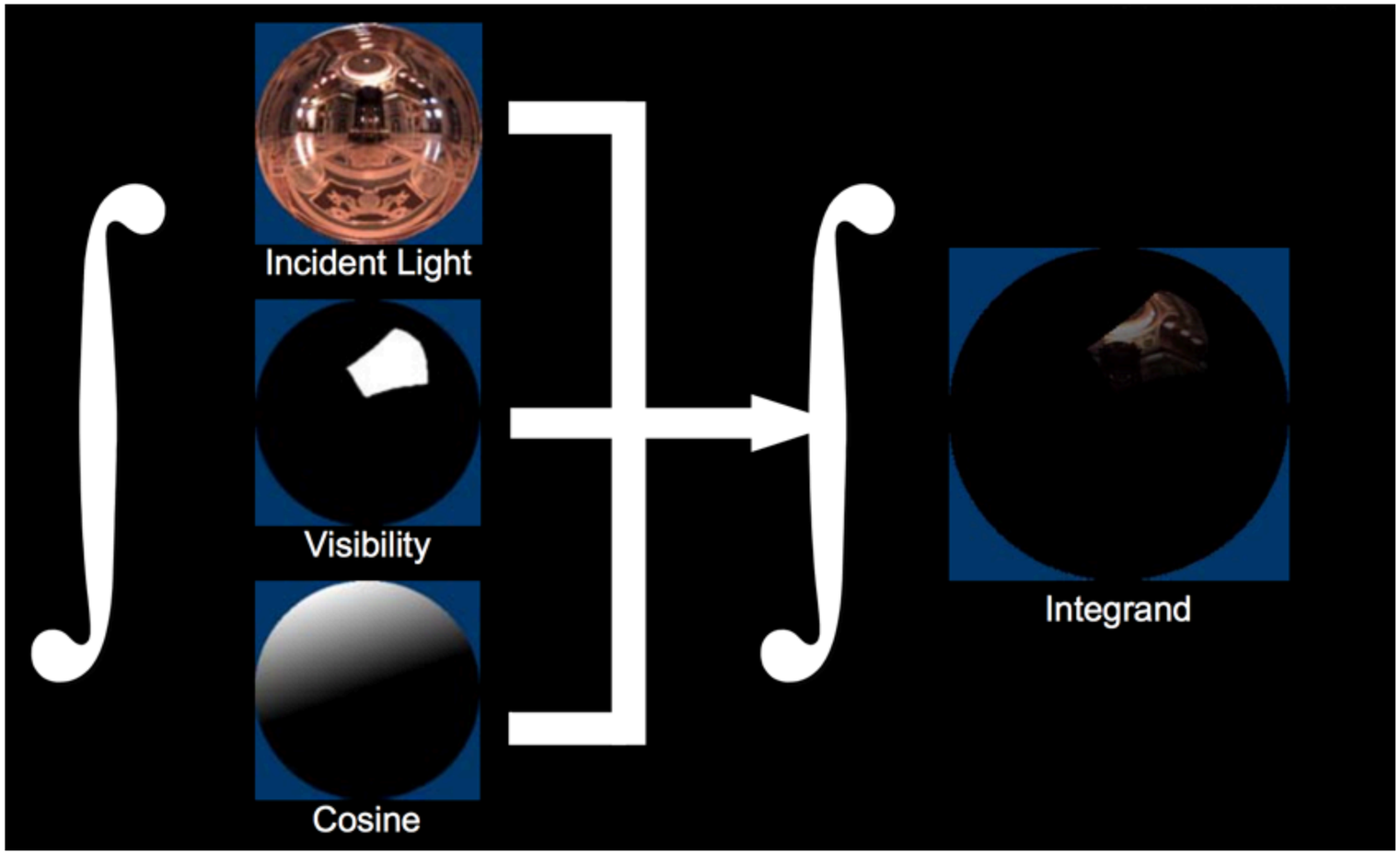
- **Grand challenge question in computer graphics:**

"Is there an O(m log m) spectral transform for general meshes?"

- **Provide a positive, constructive answer; get an A in this class.**
- **...Also revolutionize computational science (again).**

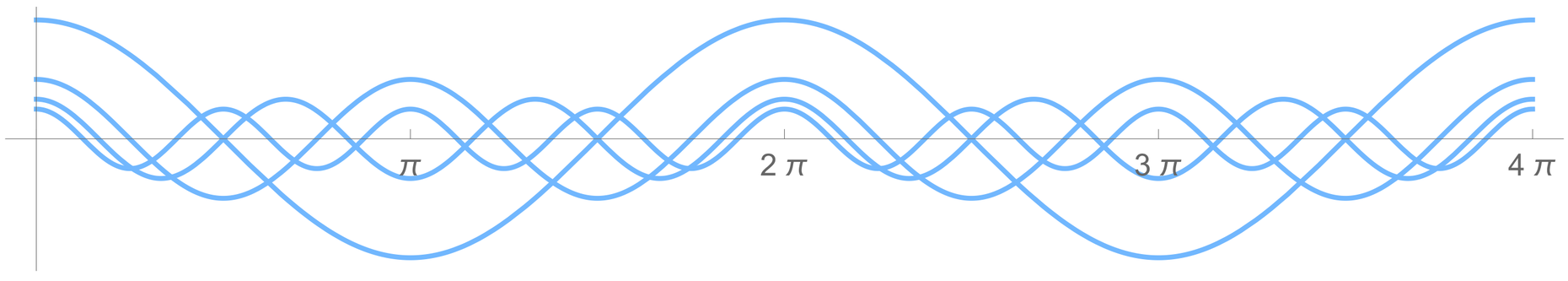
Cool Example: PRT

- "Precomputed radiance transfer"
- Diffuse illumination is convolution of illumination, visibility (& cosine term)
- So, express lights, visibility using spherical harmonics
- Fast O(m log m) evaluation of environment lighting, shadowing, diffuse interreflection, ...



Fourier Transform - Summary

- Many ways to talk about a signal
- Frequencies often make things fast/easy/…
- We looked at one particular angle (Laplacian)
- But there is much more to say:
  - completely ignored *phase* (complex numbers)
  - didn't discuss symmetry (representation theory)
  - more importantly: *didn't show how FFT works!*
- Overall, spectral viewpoint is a *gold mine* for graphics/computation.



Next time: Color & Color Spaces

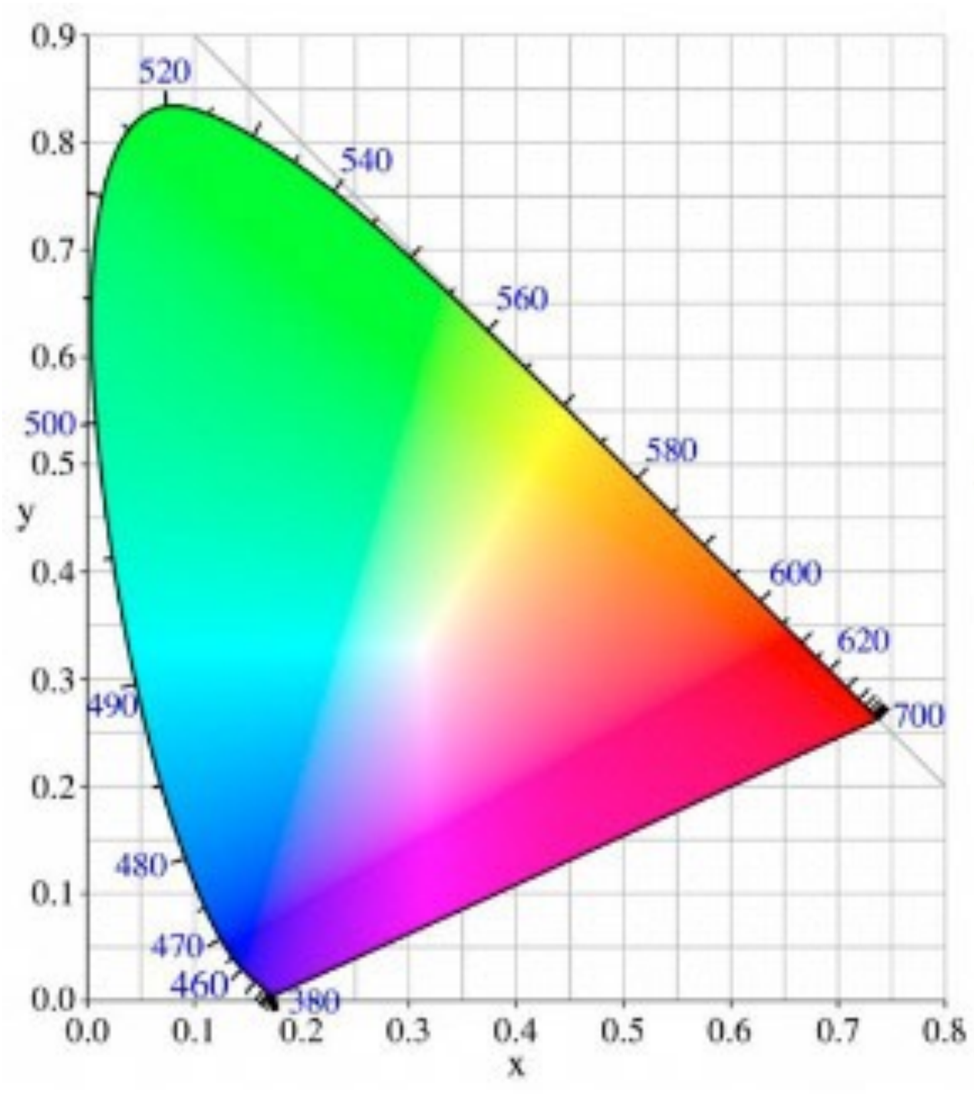
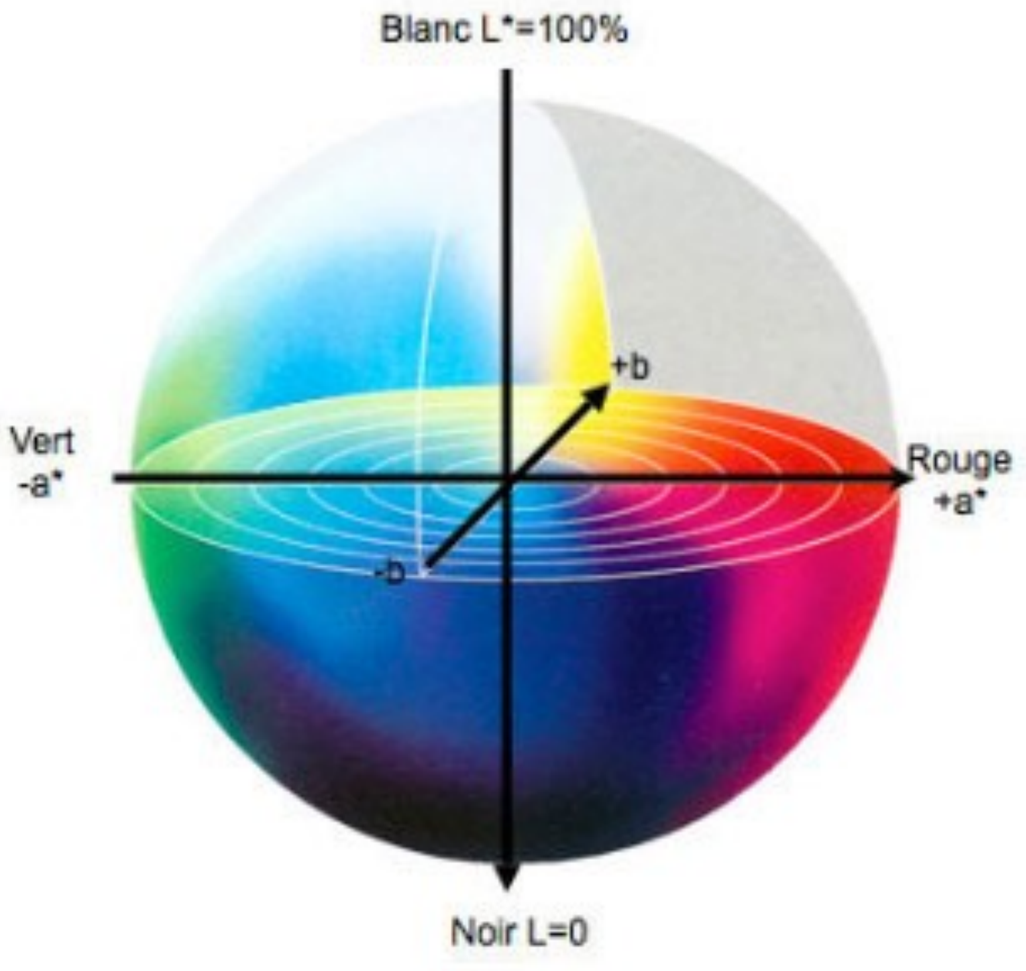


Diagram of chromaticity



Lab color space