Math 994-002
Applied and Computational Harmonic Analysis

Tuesday and Thursday, 1:00pm - 2:20pm
Wells Hall C117
Matthew Hirn

Compressed Syllabus

- **Office hours:** MTWR, 4:00 - 5:00pm
- **Book:** *A Wavelet Tour of Signal Processing*, 3rd edition, by Stéphane Mallat
- **Grading:** Homework (80%), attendance (20%), no final
- **Tentative course outline:** Fourier analysis (ch 2); discrete signal processing (ch 3); time frequency analysis (ch 4); wavelet zoom (ch 6); frames (ch 5); wavelet bases (ch 7); approximation in bases (ch 9)
- **Webpage:** https://matthewhirn.com/teaching/spring-2020-mth-994-002/

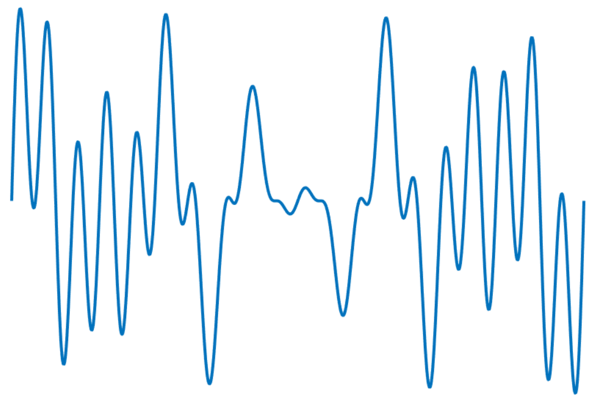
What is Harmonic Analysis?

Harmonic analysis traces its roots back to Fourier analysis...

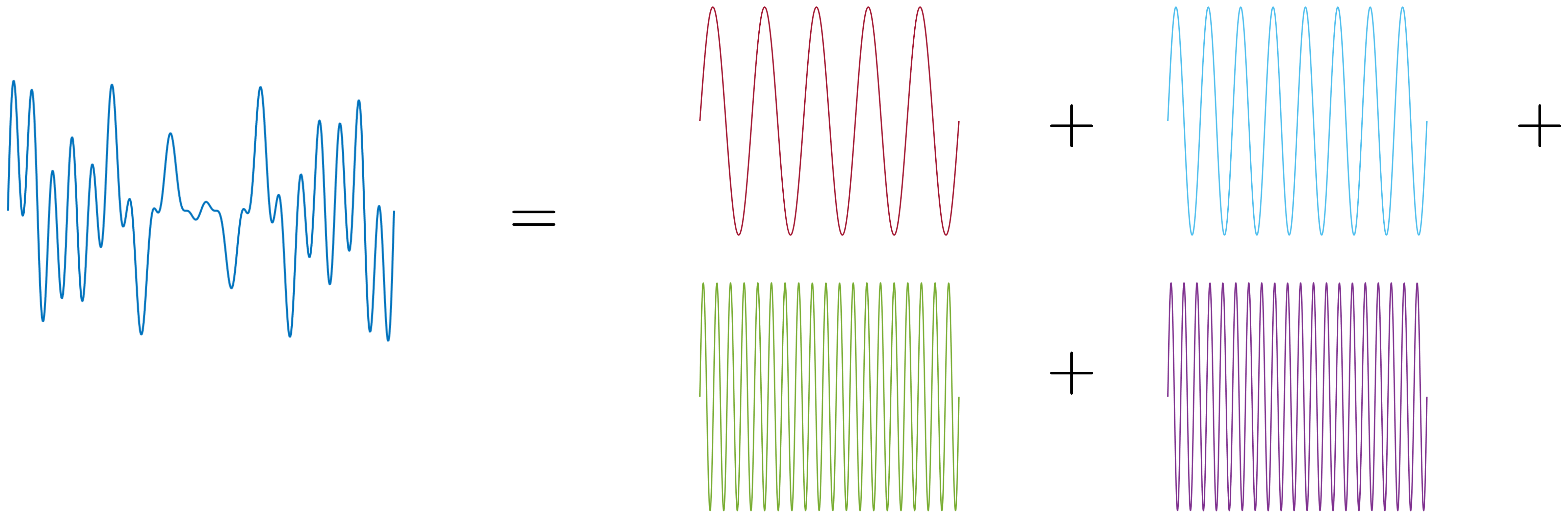
Fourier analysis goes back to the 1820's and Joseph Fourier, and his work *Théorie analytique de la chaleur*

Fourier Analysis

Fourier analysis is the study of how to decompose functions or signals like this

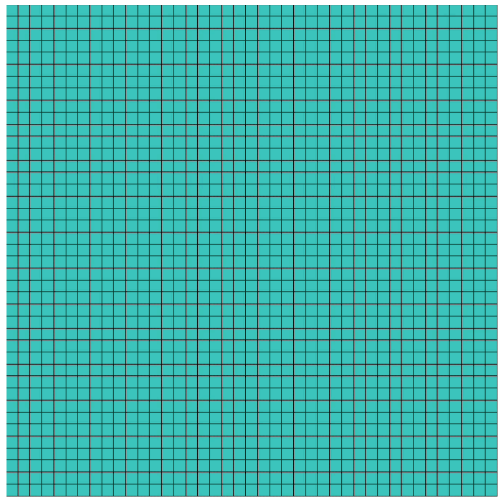


In terms of functions like this



Heat and Wave Equations

This is useful for studying heat diffusion (Fourier's original motivation)...



$$\partial_t u = \Delta u$$

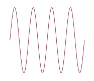
...as well as waves (such as in music)

$$\partial_t^2 u = \Delta u$$

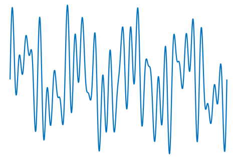
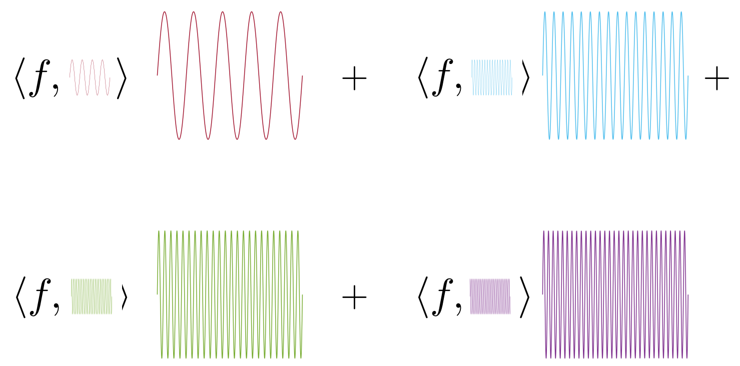
But harmonic analysis is useful for a lot more as well!

Fourier Transform

The Fourier transform for functions is:

$$\widehat{f}(\omega) = \int_{-\infty}^{+\infty} f(x)e^{-ix\omega}\,dx = \langle f,$$  $\rangle$

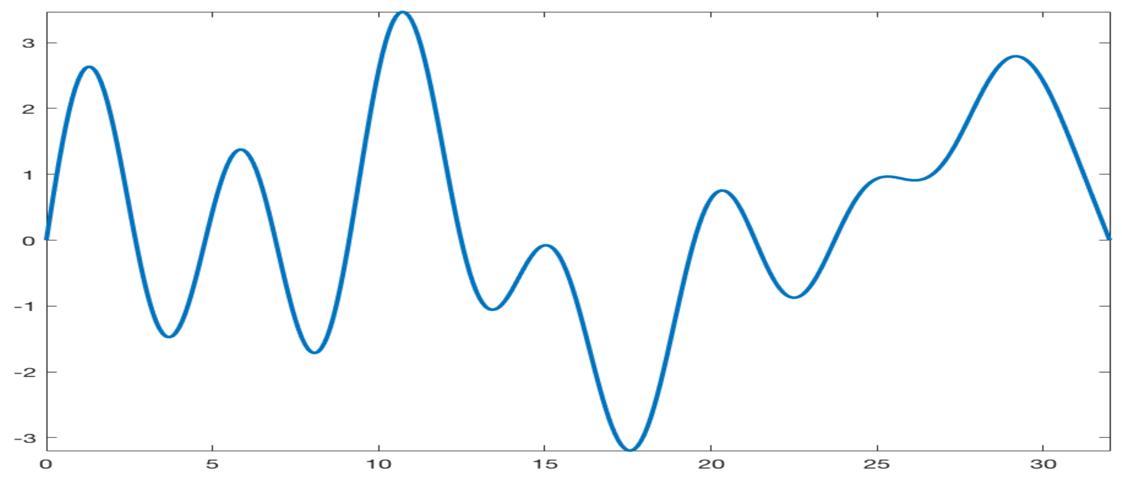
The Fourier inversion formula is:

 $=$ $$f(x) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \widehat{f}(\omega)e^{ix\omega}\,d\omega$$ $=$ 

The Fourier transform is like computing basis coefficients of $f(x)$ in the "ONB" $\mathcal{B} = \{e^{ix\omega}\}_{\omega\in\mathbb{R}}$, and the Fourier inversion formula is analogous to an ONB reconstruction

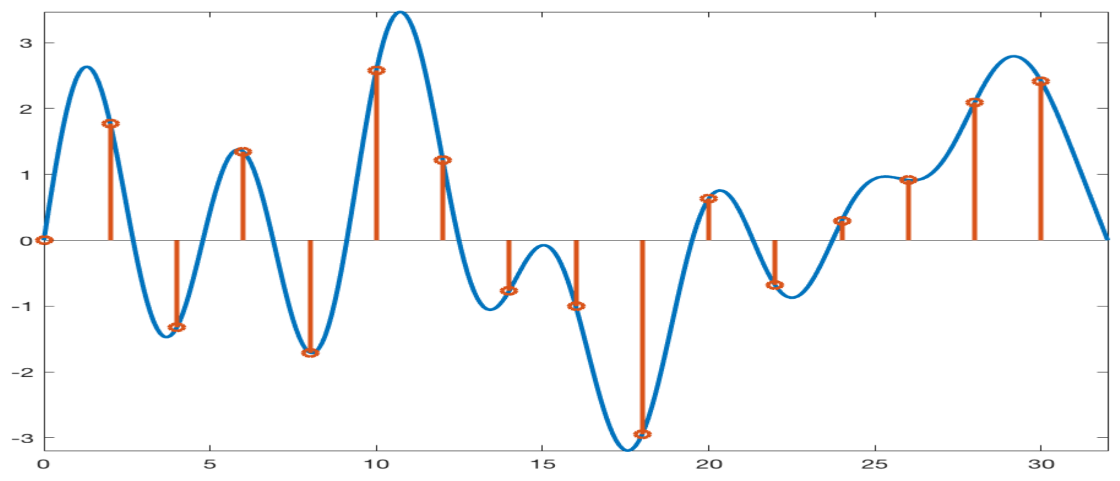
Signal Processing: Analog to Digital

On a computer, we only store samples of a signal. How many do we need?



Signal Processing: Analog to Digital

On a computer, we only store samples of a signal. How many do we need?



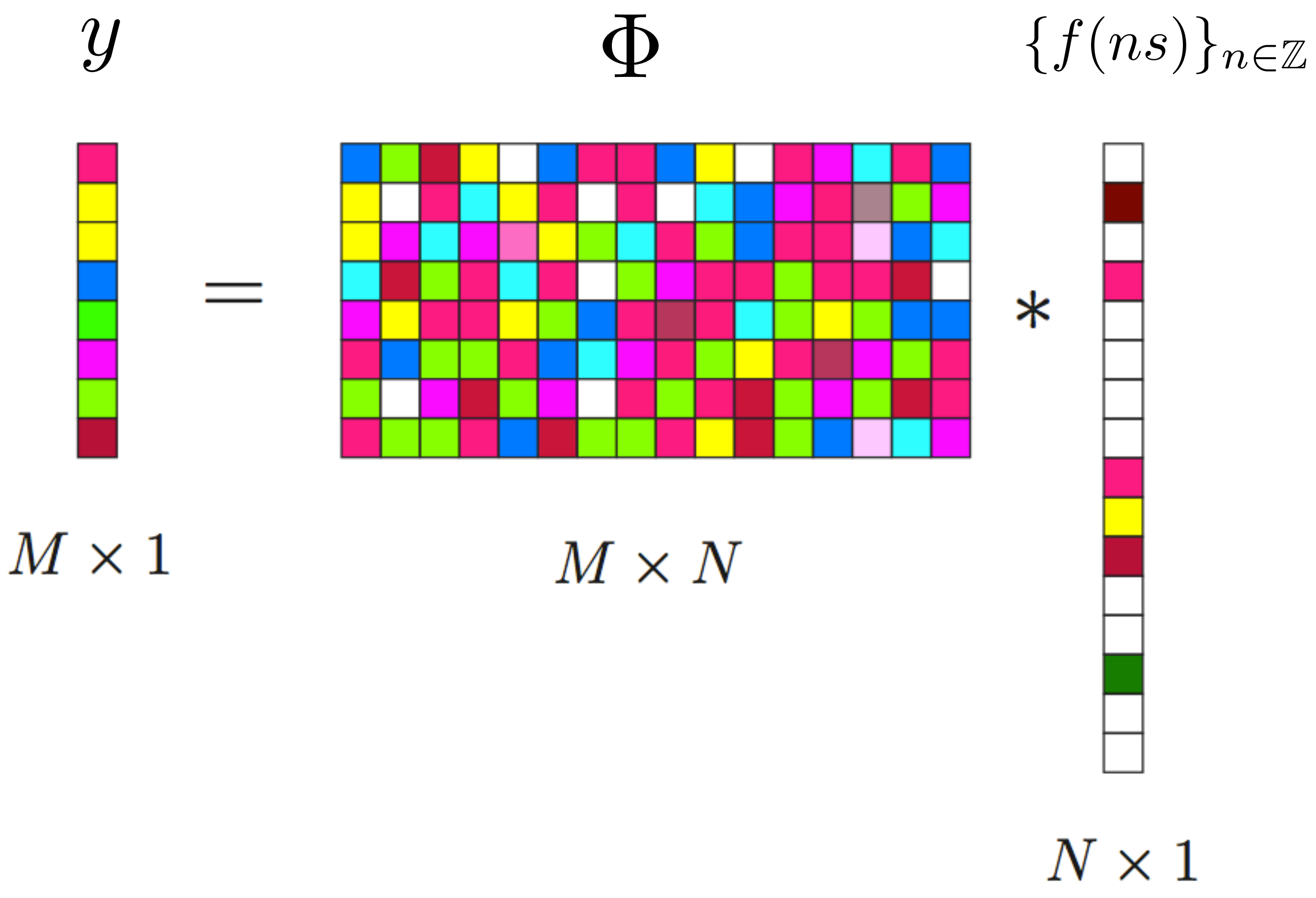
Nyquist/Shannon/Whittaker: If $\widehat{f}(\omega) = 0$ for all $\omega \notin [-\pi/s, \pi/s]$, then we need only need $\{f(ns)\}_{n\in\mathbb{Z}}$. More precisely,

$$f(x) = \sum_{n\in\mathbb{Z}} f(ns)\phi_s(x - ns)$$

with

$$\phi_s(x) = \frac{\sin(\pi x/s)}{\pi x/s}$$

Compressive Sensing



If the samples $\{f(ns)\}_{n\in\mathbb{Z}}$ are sparse, and if we only care about being able to reconstruct $f(x)$ 99.99% of the time, we can get away with even less samples, far fewer in fact!

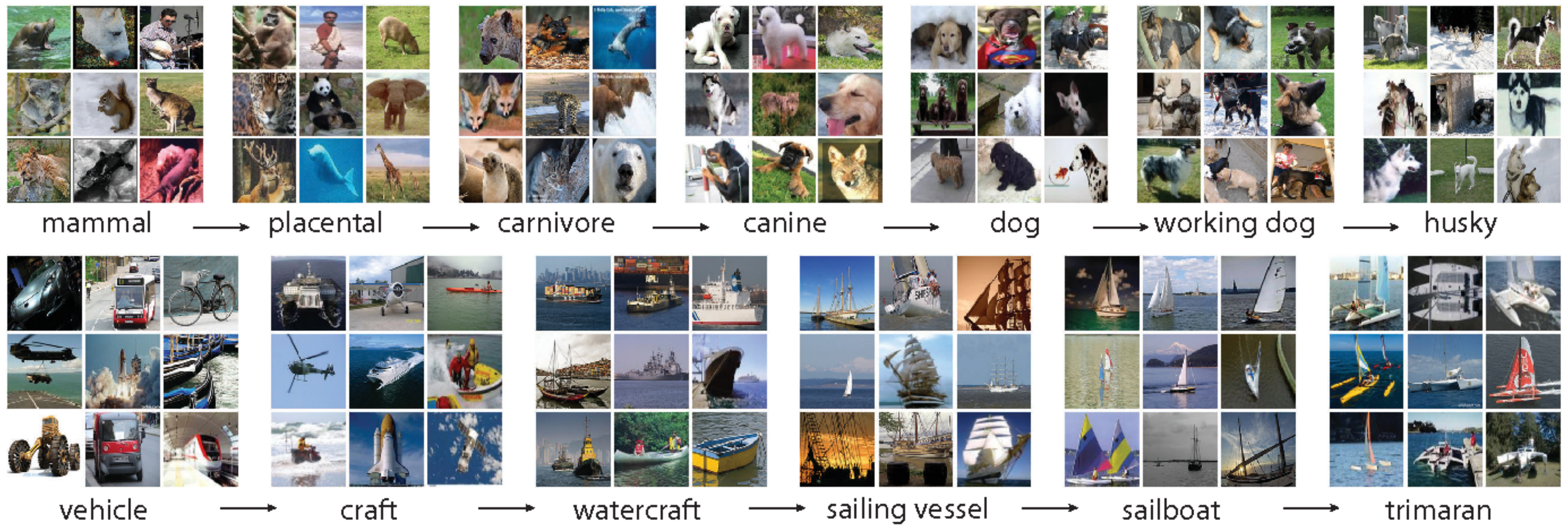
Fast Calculations

That's great, but how fast can we compute things like the Fourier transform after we sample the signal?

$$\widehat{f}(\omega) = \int_{-\infty}^{+\infty} f(x)e^{-ix\omega}\,dx$$

Answer: Fast, via the "Fast Fourier Transform!" Much faster than one might initially think in fact.

Image Processing



If we have a lot of images, we may want to compress them. This requires a sparse representation

Fourier analysis alone will not work, because images often have sharp transitions and other fine scale structures, in addition to macroscopic patterns

We need something multiscale

Image Processing

A wavelet ψ is a localized waveform with zero average

In 2D, we dilate and rotate ψ,

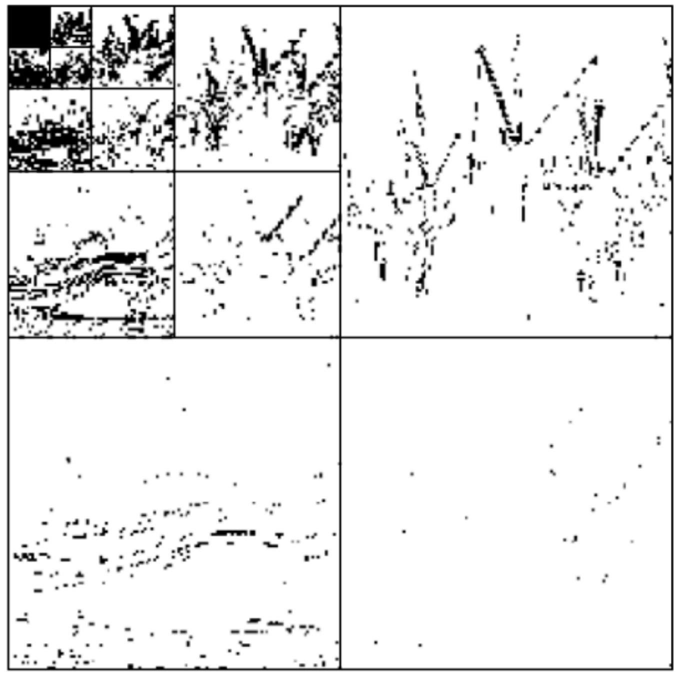
$$\psi_{j,\theta}(u) = 2^{-2j}\psi(2^{-j}R_\theta^{-1}u)$$

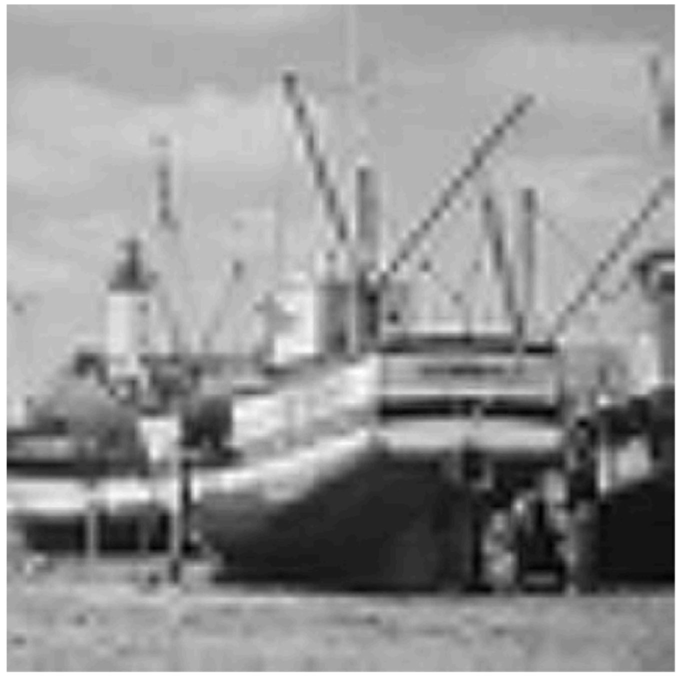
and test the image $x(u)$ against $\psi_{j,\theta}$ by computing the wavelet transform:

$$Wx = \{x * \psi_{j,\theta}(u)\}_{j,\theta,u}$$

Linear and nonlinear image compression

(a)


(b)


(c)

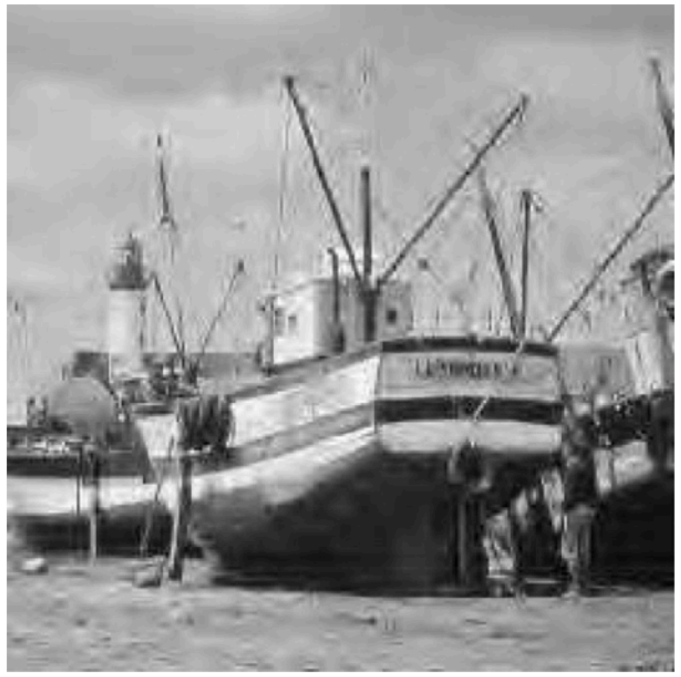

(d)

Image Processing

A wavelet ψ is a localized waveform with zero average

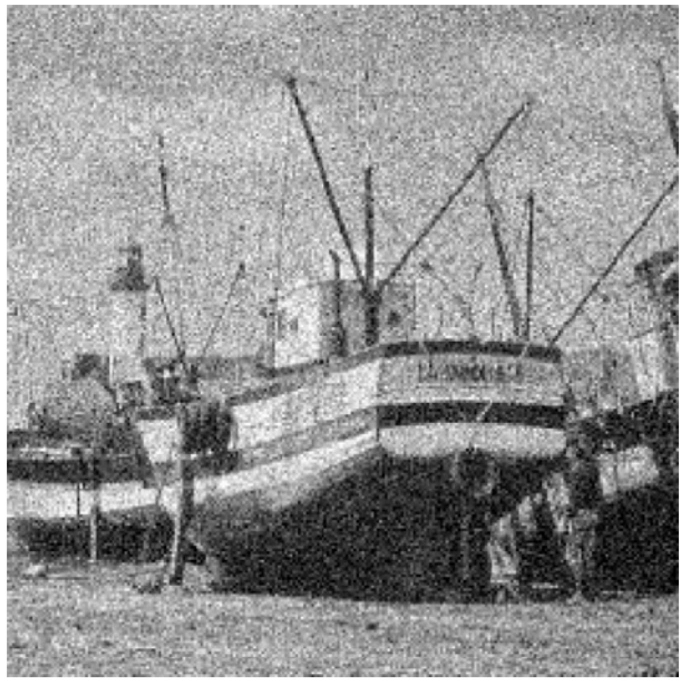
In 2D, we dilate and rotate ψ,

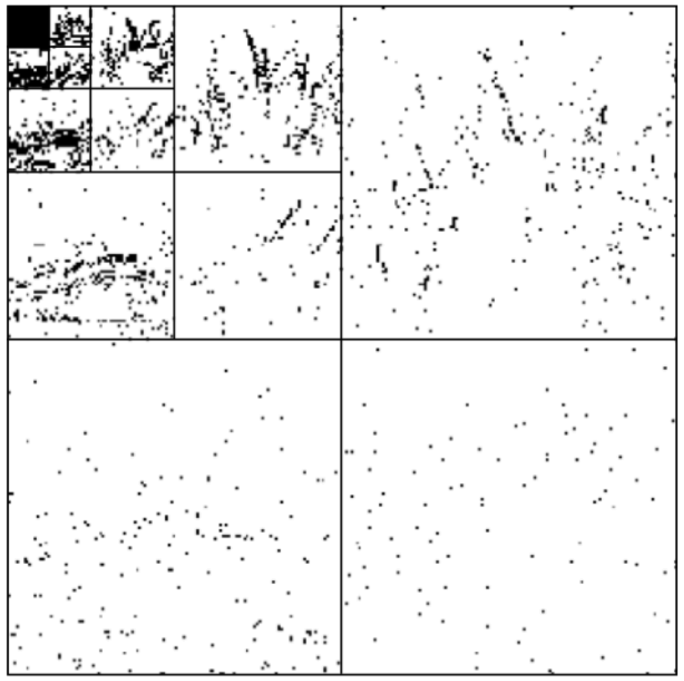
$$\psi_{j,\theta}(u) = 2^{-2j}\psi(2^{-j}R_\theta^{-1}u)$$

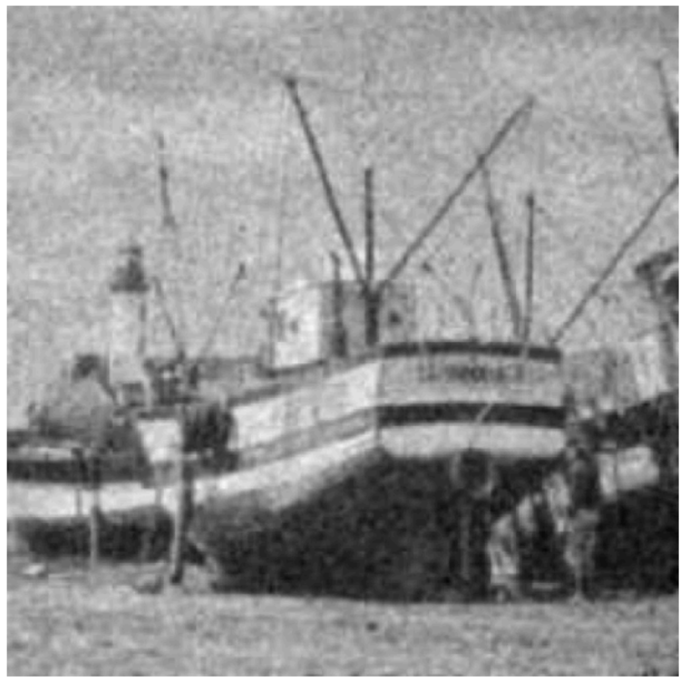
and test the image $x(u)$ against $\psi_{j,\theta}$ by computing the wavelet transform:

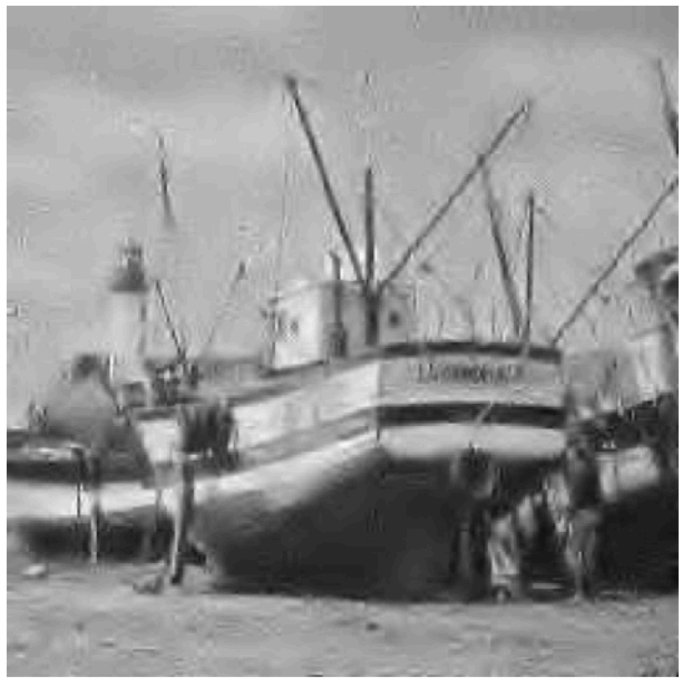
$$Wx = \{x * \psi_{j,\theta}(u)\}_{j,\theta,u}$$

Linear and nonlinear image denoising


(a)

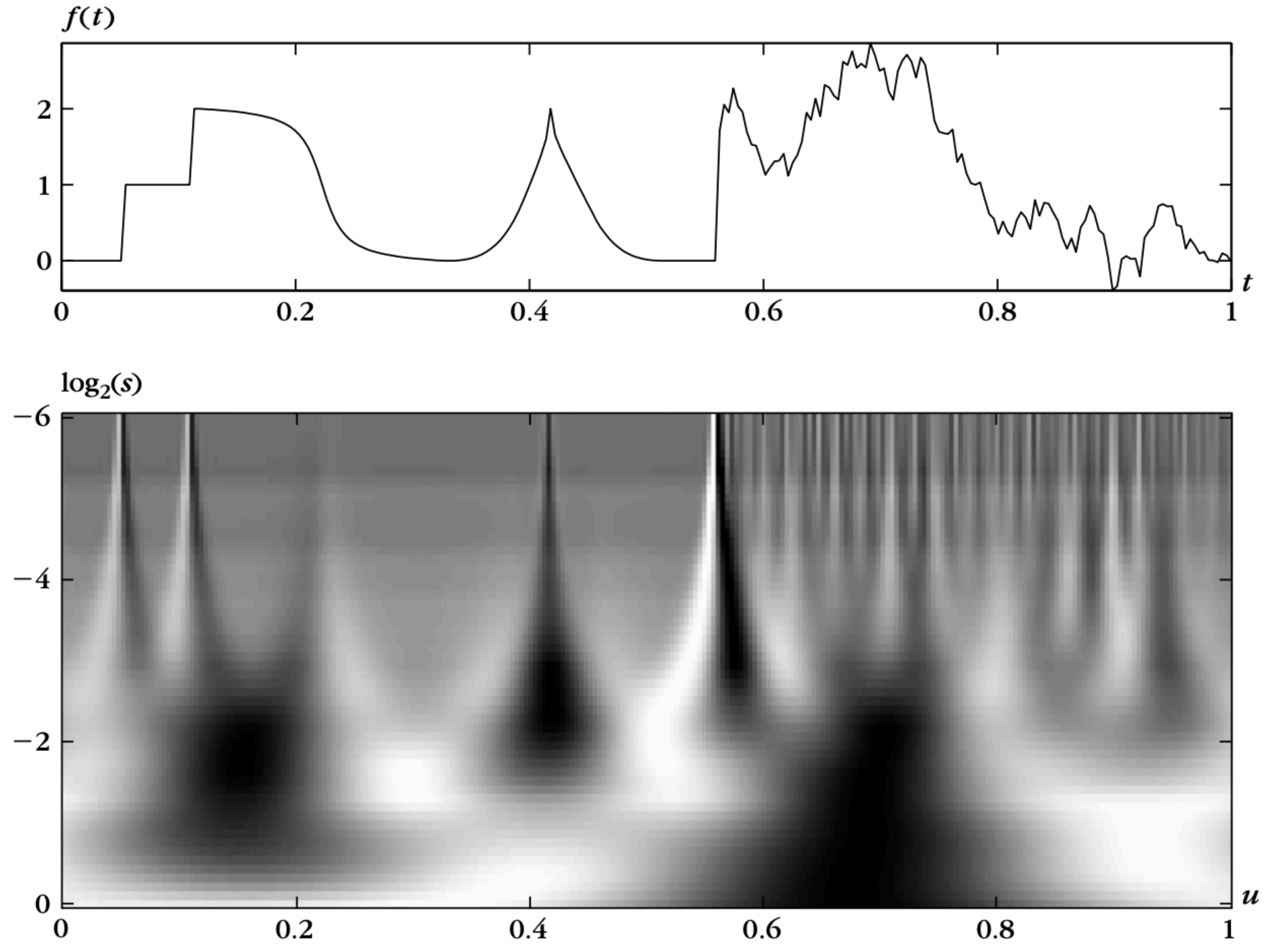

(b)


(c)


(d)

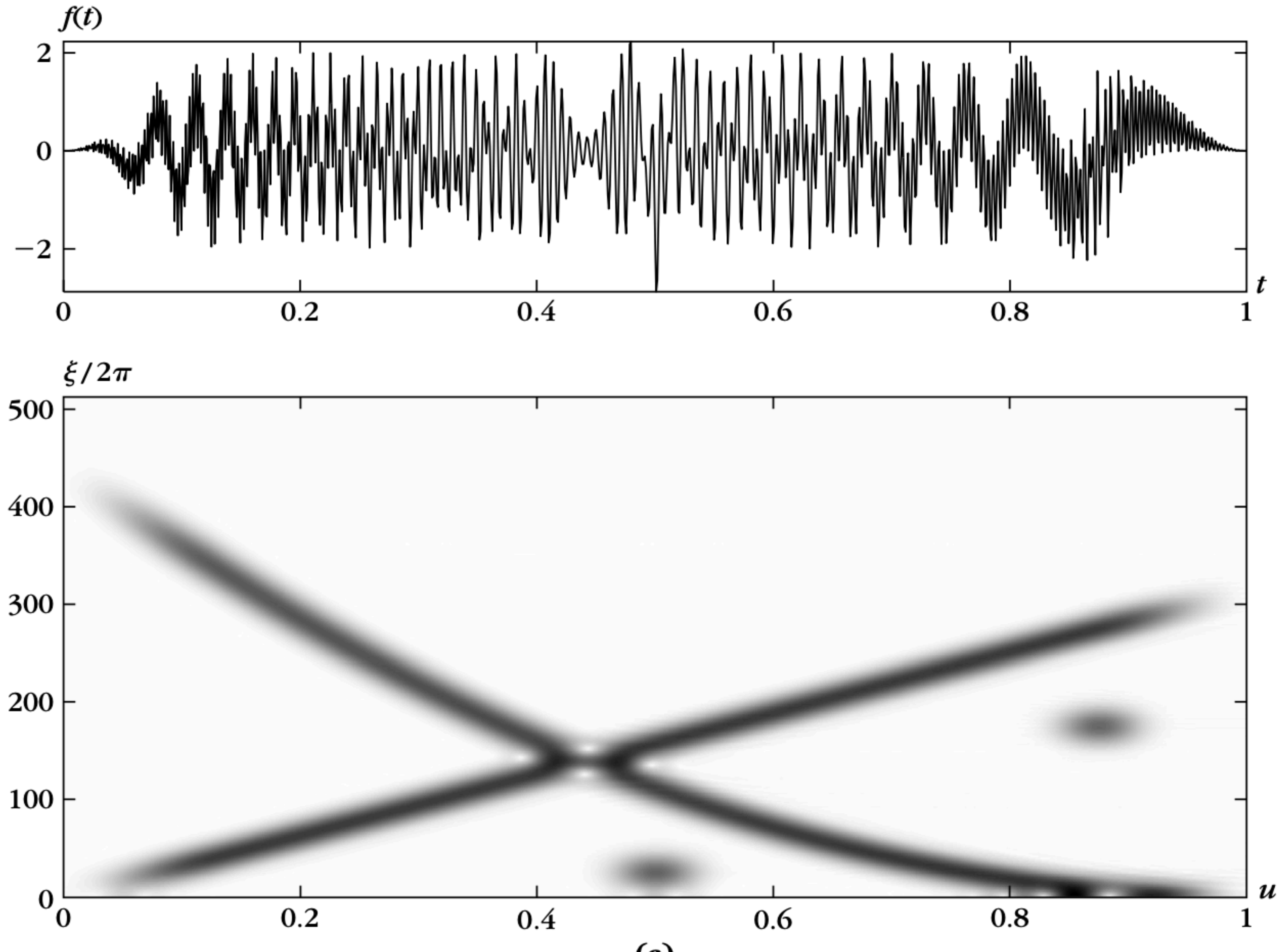
Signal Characterization

To understand the previous results, we must understand how wavelets characterize certain classes of functions



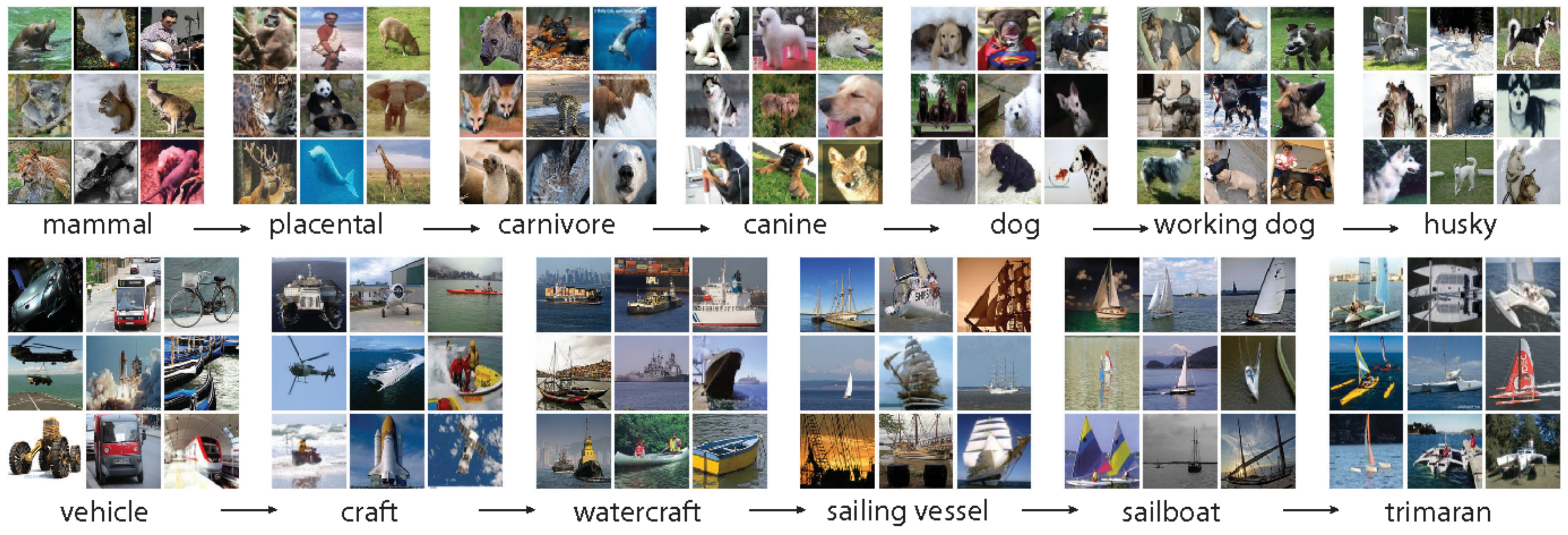
Extracting Information from Complex Signals

Sometimes we want to separate out the constituent components of complex signals. Time-frequency transforms can help with this

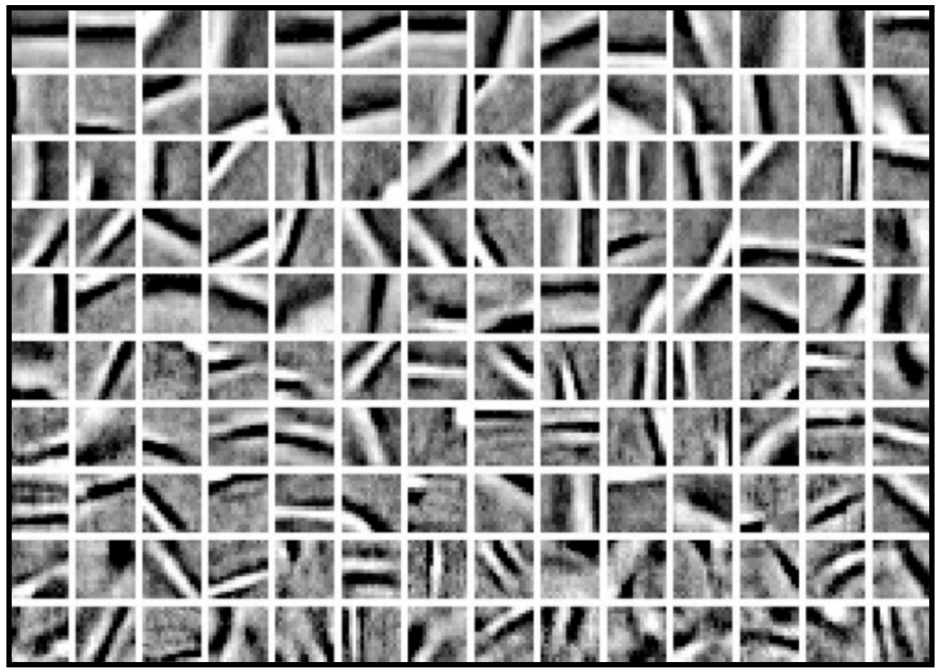


(a)

Dictionary Learning



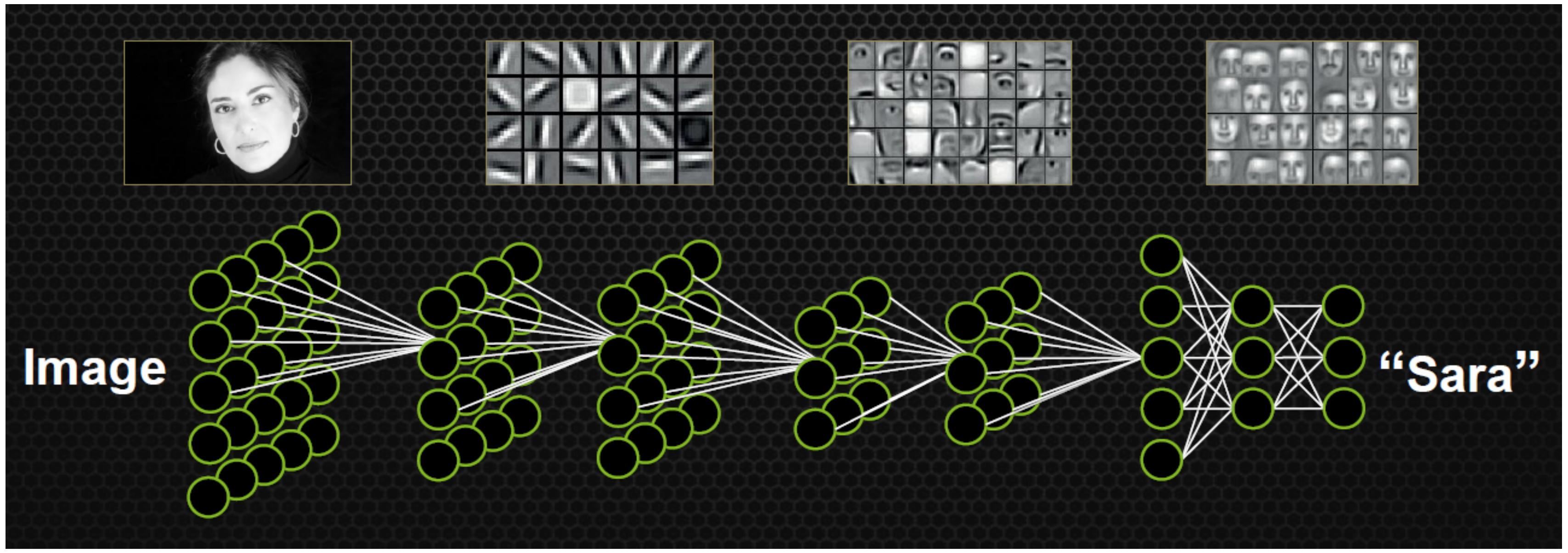
Rather than constructing wavelets, can we learn a dictionary of waveforms that gives a sparse representation of natural images?

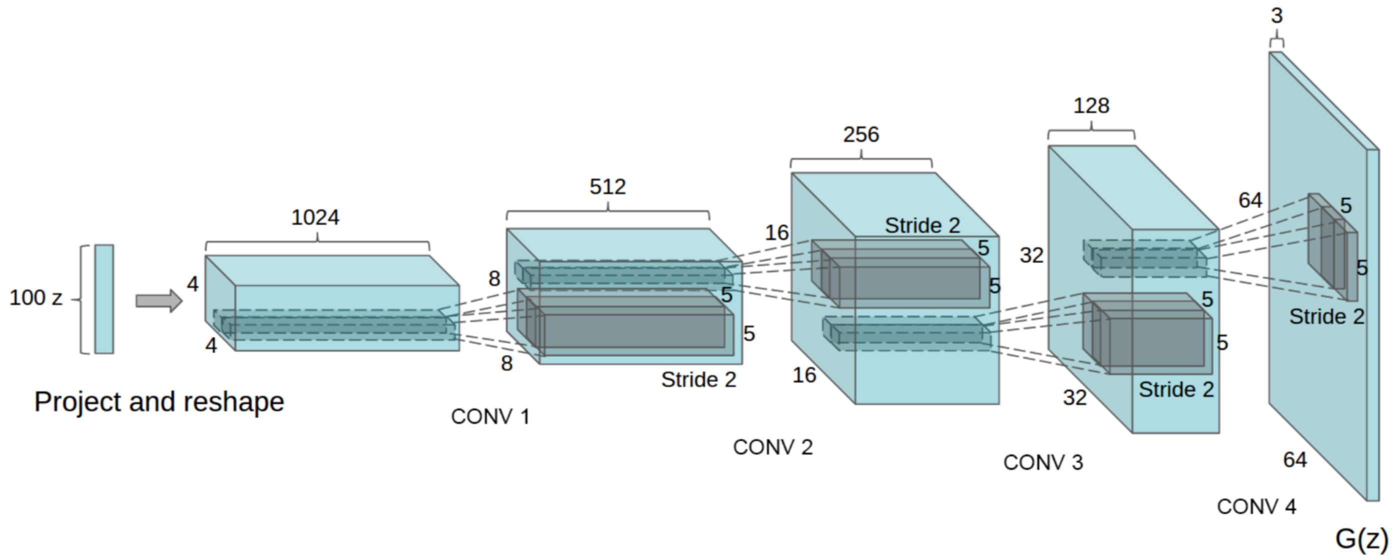


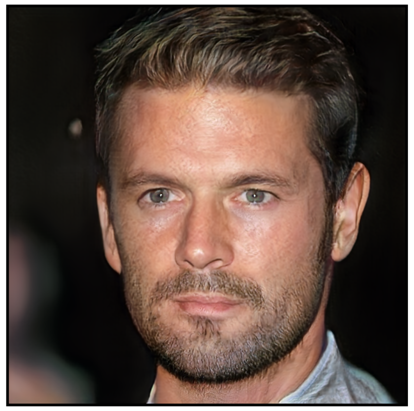
Machine learned waveforms that give a sparse representation of natural images (they look very similar to wavelets!)

Deep Learning

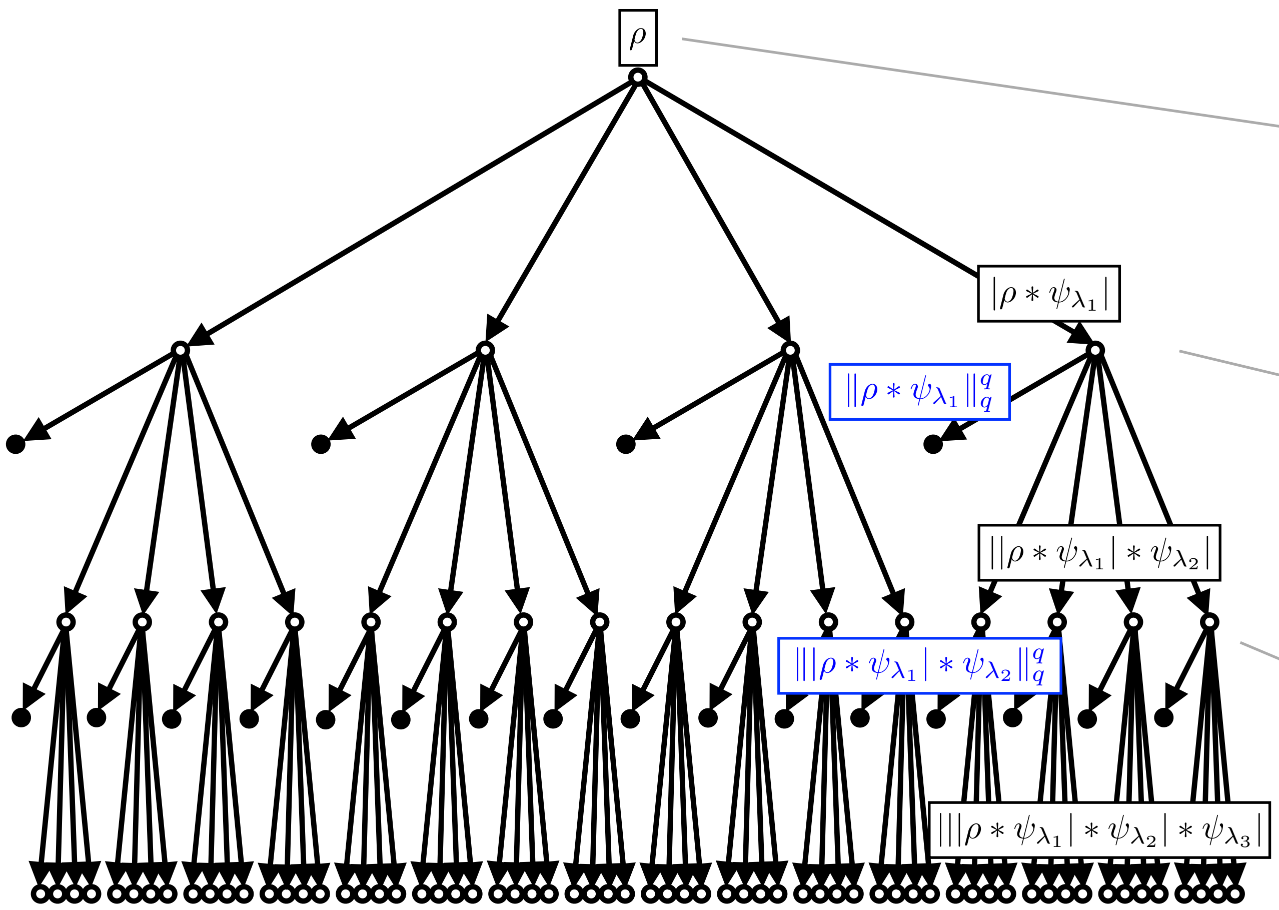
Of course we may not want to compress an image. We may want classify it or even generate new synthetic images. Deep learning, and in particular convolutional neural networks, are very good at these tasks and getting better every month

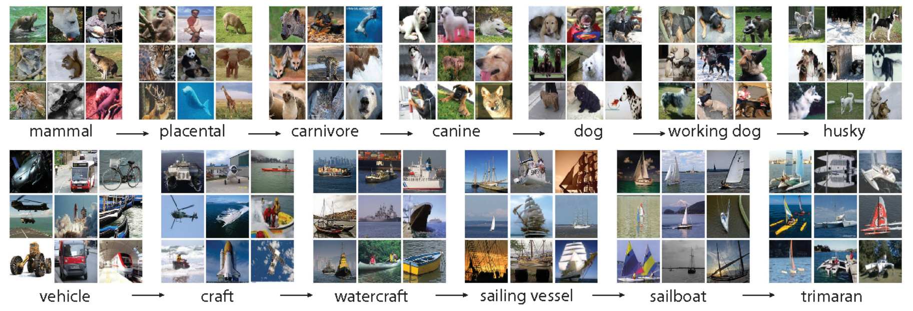


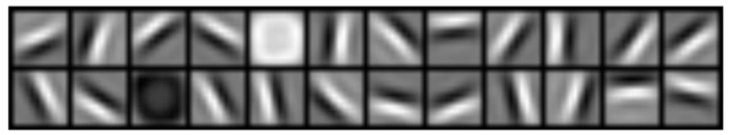


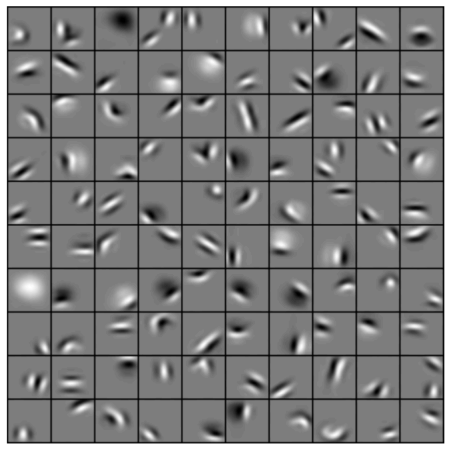


Deep Learning and Harmonic Analysis

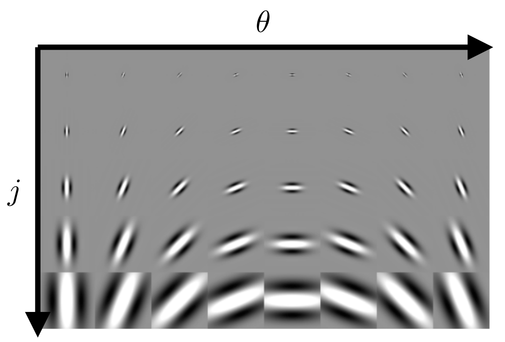

ρ
$|\rho * \psi_{\lambda_1}|$
$\|\rho * \psi_{\lambda_1}\|_q^q$
$||\rho * \psi_{\lambda_1}| * \psi_{\lambda_2}|$
$\||\rho * \psi_{\lambda_1}| * \psi_{\lambda_2}\|_q^q$
$|||\rho * \psi_{\lambda_1}| * \psi_{\lambda_2}| * \psi_{\lambda_3}|$



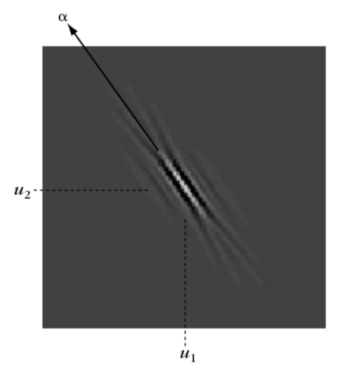




Machine learned filters at each layer (again similar to wavelets!)



Gabor wavelets $\psi_{j,\theta}$



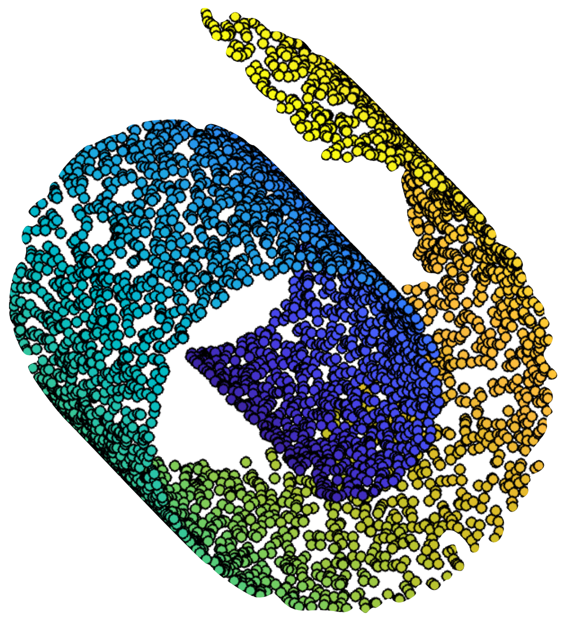
Curvelet $\psi_{j,\alpha}$

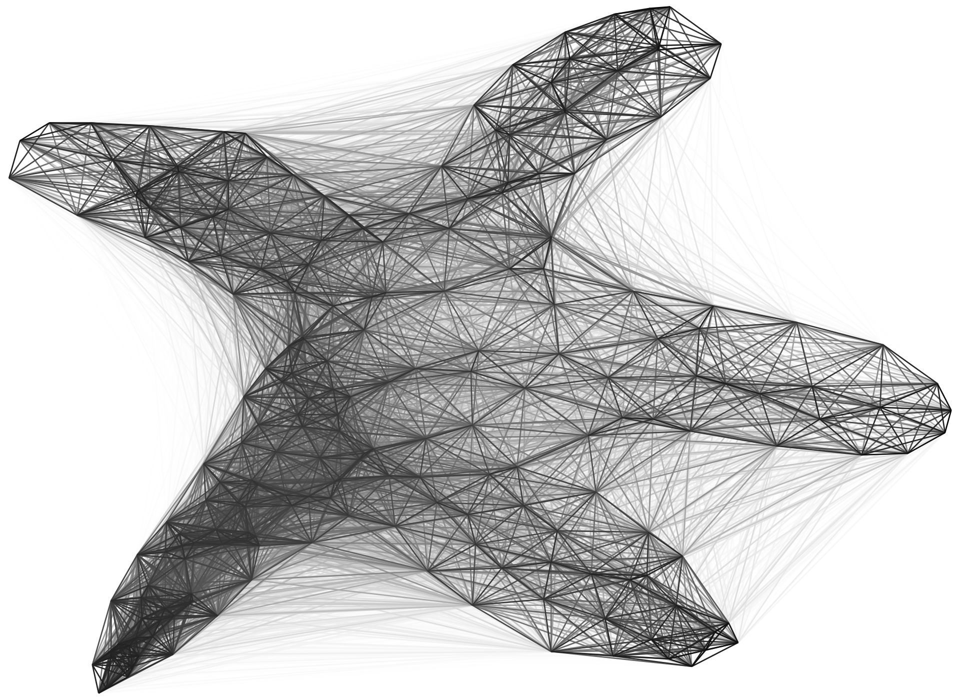
We can use tools from harmonic analysis to understand why deep learning works, and even better, to inform us how to push beyond the current state of the art

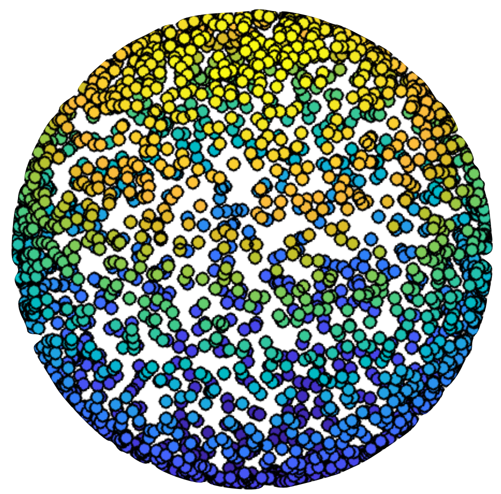
Harmonic Analysis on Graphs and Manifolds

In unsupervised learning tasks, we often want to organize high dimensional data in some fashion, e.g., cluster it, or find a low dimensional representation of the data

It turns out we can do this by understanding how to do harmonic analysis on graphs and manifolds, like these:







Harmonic Analysis on Graphs and Manifolds

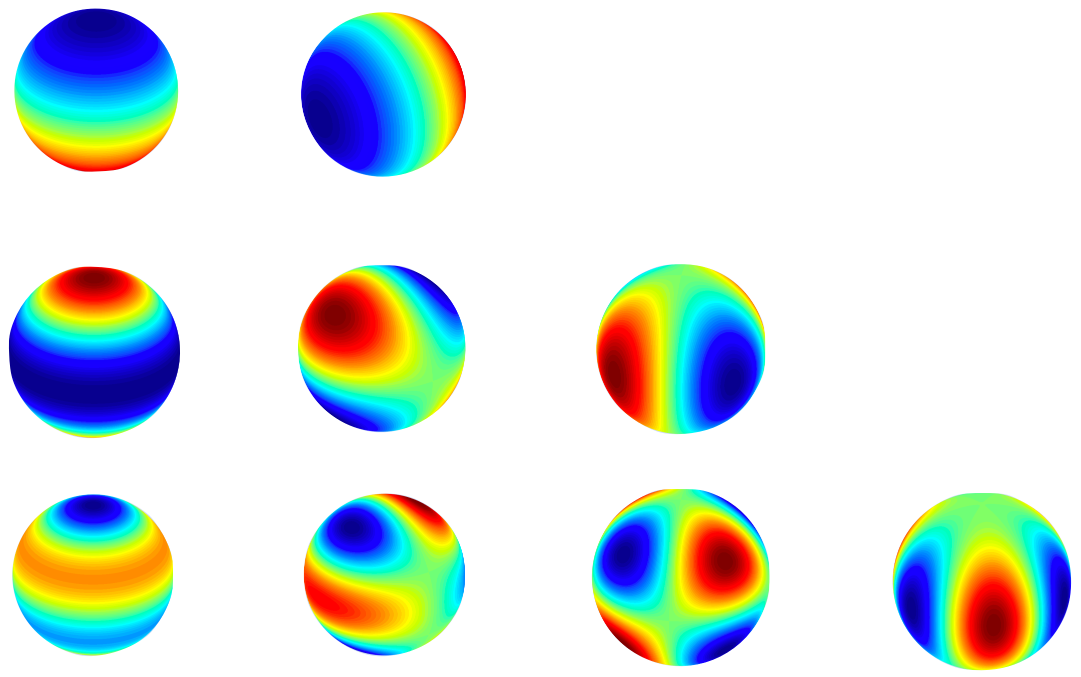
What are the harmonics on manifolds?

Note that $\Delta e^{ix\cdot\omega} = -|\omega|^2 e^{ix\cdot\omega}$

Thus $e^{ix\cdot\omega}$ is eigenfunction of Δ with eigenvalue $-|\omega|^2$

Let $\mathcal{M}$ be a compact Riemannian manifold. It turns out we can define Δ for functions $f : \mathcal{M} \to \mathbb{C}$. The eigenfunctions of Δ define the harmonics of $\mathcal{M}$

Similarly let $G = (V, E, W)$ be a weighted graph. In this case we define an analog of Δ, which is the graph Laplacian $\mathbf{L}$. Its eigenvectors are the harmonics of G

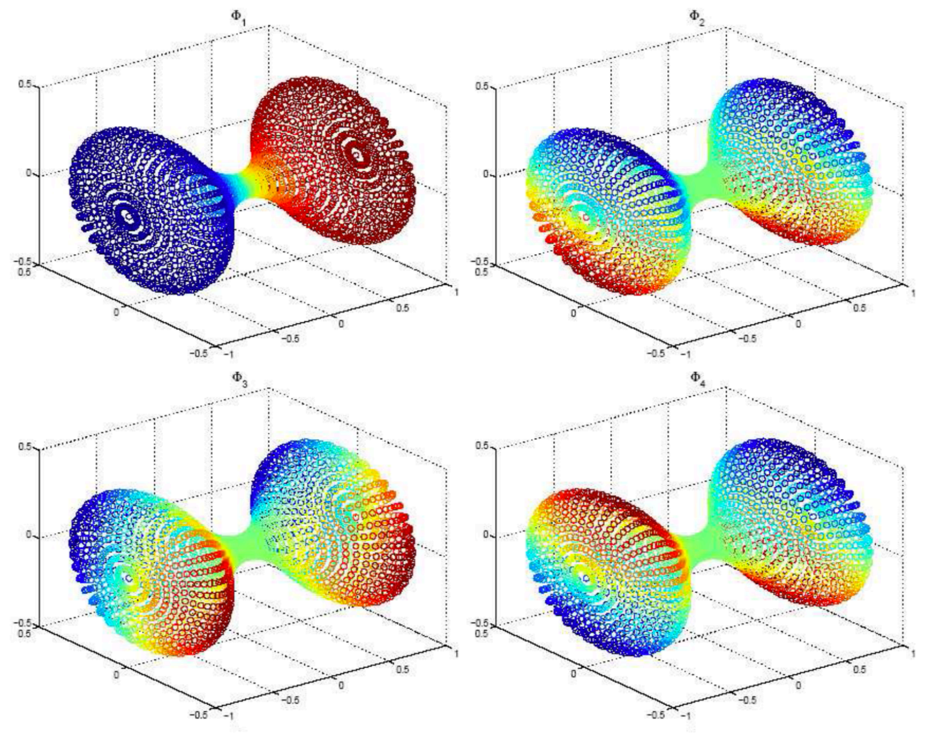


Manifold Learning

We can turn any dataset into a weighted graph if we can measure some notion of similarity between data points

Once we have a graph, we can construct the graph Laplacian

These eigenvectors can be used to cluster the data and/or give low dimensional embeddings of the data


Φ_1
Φ_2
Φ_3
Φ_4