Fourier Series And The Fourier Transform

Et Applications

Nicholas Anthony Giannoulis

School of Mathematics and Statistics
University of Sydney

Summary

Fourier Series

Signals

- Define a signal to be a real valued (Riemann) integrable function on some time interval T.
- $f : T \to \mathbb{R}$, $T = [a, b]$, $a, b \in \mathbb{R}$
- $\forall [t_0, t_1] \subseteq T$, $\left| \int_{t_0}^{t_1} f(t)dt \right| < \infty$

Signals

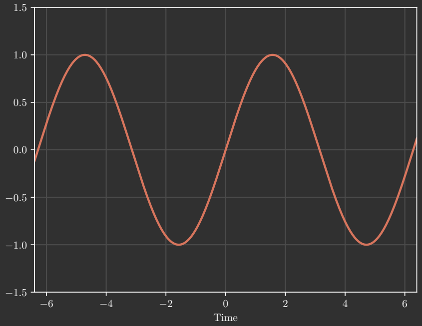
- Define a signal to be a real valued (Riemann) integrable function on some time interval T.
- $f : T \to \mathbb{R},\ T = [a, b],\ a, b \in \mathbb{R}$
- $\forall [t_0, t_1] \subseteq T,\ \left|\int_{t_0}^{t_1} f(t)dt\right| < \infty$

Signals

- Define a signal to be a real valued (Riemann) integrable function on some time interval T.
- $f : T \to \mathbb{R}$, $T = [a, b]$, $a, b \in \mathbb{R}$
- $\forall [t_0, t_1] \subseteq T$, $\left| \int_{t_0}^{t_1} f(t) dt \right| < \infty$

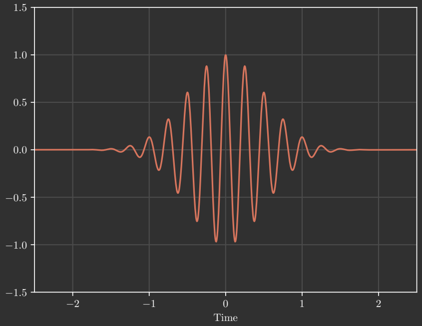
Signals

$$f(t) = \sin(t)$$



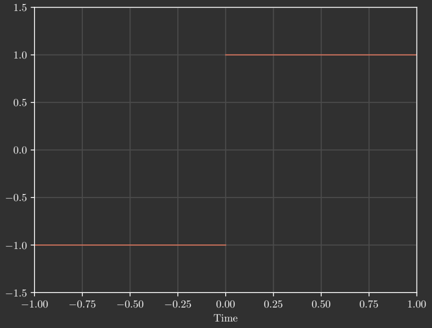
Signals

$$f(t) = \cos(25x)e^{-2x^2}$$



Signals

$$f(t) = \begin{cases} 1 & t \geq 0 \\ -1 & t < 0 \end{cases}$$



Fourier Series

- In the interest of brevity we will limit our study of signals to those defined on the interval $T = [0, 1]$
- All such signals can be expressed as a (possibly infinite) sum of sinusoidal signals with integer valued frequencies;

$$f(t) \sim \sum_{n \in \mathbb{Z}} \hat{f}(n) e^{2\pi int}$$

$$\hat{f}(n) := \int_0^1 f(t) e^{-2\pi int} dt$$

$$\lim_{n \to \pm\infty} \hat{f}(n) = 0$$

Fourier Series

- In the interest of brevity we will limit our study of signals to those defined on the interval $T = [0, 1]$
- All such signals can be expressed as a (possibly infinite) sum of sinusoidal signals with integer valued frequencies;

$$f(t) \sim \sum_{n \in \mathbb{Z}} \hat{f}(n) e^{2\pi i n t}$$

$$\hat{f}(n) := \int_0^1 f(t) e^{-2\pi i n t} dt$$

$$\lim_{n \to \pm\infty} \hat{f}(n) = 0$$

Fourier Series

- In the interest of brevity we will limit our study of signals to those defined on the interval $T = [0, 1]$
- All such signals can be expressed as a (possibly infinite) sum of sinusoidal signals with integer valued frequencies;

$$f(t) \sim \sum_{n\in\mathbb{Z}} \hat{f}(n)e^{2\pi int}$$

$$\hat{f}(n) := \int_0^1 f(t)e^{-2\pi int}dt$$

$$\lim_{n\to\pm\infty} \hat{f}(n) = 0$$

Fourier Series

- In the interest of brevity we will limit our study of signals to those defined on the interval $T = [0,1]$
- All such signals can be expressed as a (possibly infinite) sum of sinusoidal signals with integer valued frequencies;

$$f(t) \sim \sum_{n\in\mathbb{Z}} \hat{f}(n) e^{2\pi i n t}$$

$$\hat{f}(n) := \int_0^1 f(t) e^{-2\pi i n t} dt$$

$$\lim_{n\to\pm\infty} \hat{f}(n) = 0$$

Fourier Series

- In the interest of brevity we will limit our study of signals to those defined on the interval $T = [0, 1]$
- All such signals can be expressed as a (possibly infinite) sum of sinusoidal signals with integer valued frequencies;

$$f(t) \sim \sum_{n \in \mathbb{Z}} \hat{f}(n) e^{2\pi i n t}$$

$$\hat{f}(n) := \int_0^1 f(t) e^{-2\pi i n t} dt$$

$$\lim_{n \to \pm\infty} \hat{f}(n) = 0$$

Fourier Series

- Since we specified signals to be real valued functions, their Fourier series admit the following form.

$$f(t) \sim -\frac{a_0}{2} + \sum_{n\in\mathbb{N}} a_n \cos(2\pi n t) + \sum_{n\in\mathbb{N}} a_n \sin(2\pi n t)$$

$$a_n := 2\int_0^1 f(t)\cos(2\pi n t)dt$$

$$b_n := 2\int_0^1 f(t)\sin(2\pi n t)dt$$

Fourier Series

- Since we specified signals to be real valued functions, their Fourier series admit the following form.

$$f(t) \sim -\frac{a_0}{2} + \sum_{n\in\mathbb{N}} a_n \cos(2\pi n t) + \sum_{n\in\mathbb{N}} a_n \sin(2\pi n t)$$

$$a_n := 2\int_0^1 f(t)\cos(2\pi n t)dt$$

$$b_n := 2\int_0^1 f(t)\sin(2\pi n t)dt$$

Fourier Series

- Since we specified signals to be real valued functions, their Fourier series admit the following form.

$$f(t) \sim -\frac{a_0}{2} + \sum_{n \in \mathbb{N}} a_n \cos(2\pi n t) + \sum_{n \in \mathbb{N}} a_n \sin(2\pi n t)$$

$$a_n := 2 \int_0^1 f(t) \cos(2\pi n t) dt$$

$$b_n := 2 \int_0^1 f(t) \sin(2\pi n t) dt$$

Fourier Series

- Since we specified signals to be real valued functions, their Fourier series admit the following form.

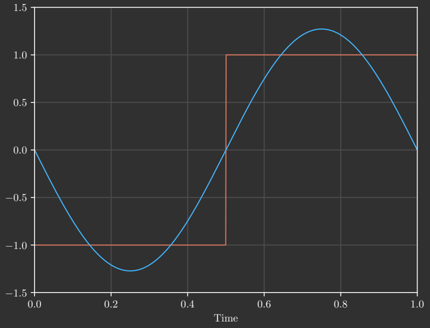
$$f(t) \sim -\frac{a_0}{2} + \sum_{n\in\mathbb{N}} a_n \cos(2\pi nt) + \sum_{n\in\mathbb{N}} a_n \sin(2\pi nt)$$

$$a_n := 2\int_0^1 f(t)\cos(2\pi nt)dt$$

$$b_n := 2\int_0^1 f(t)\sin(2\pi nt)dt$$

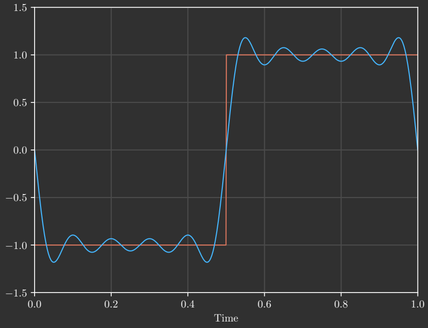
Fourier Series

$$f(t) = \begin{cases} 1 & t \geq \frac{1}{2} \\ -1 & t < \frac{1}{2} \end{cases}$$



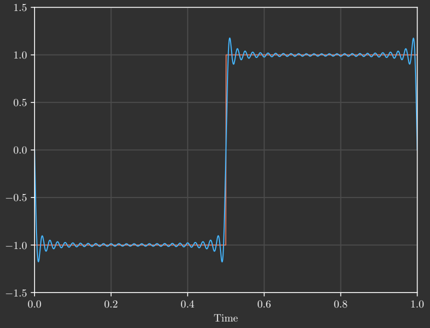
Fourier Series

$$f(t) = \begin{cases} 1 & t \geq \frac{1}{2} \\ -1 & t < \frac{1}{2} \end{cases}$$



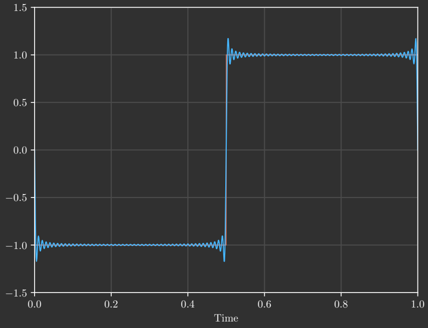
Fourier Series

$$f(t) = \begin{cases} 1 & t \geq \frac{1}{2} \\ -1 & t < \frac{1}{2} \end{cases}$$



Fourier Series

$$f(t) = \begin{cases} 1 & t \geq \frac{1}{2} \\ -1 & t < \frac{1}{2} \end{cases}$$



Fourier series on the n-cube

- Fourier analysis of this nature generalises near seamlessly to higher dimensions.
- Extend the definition of signals to real valued functions on the n-cube; $T^n = [0,1]^n$.

$$f(\bar{t}) \sim \sum_{\bar{x}\in\mathbb{N}^n} \hat{f}(\bar{x})e^{2\pi i\bar{t}}, \; \bar{x} = (x_1, x_2, \ldots, x_n)$$

$$\hat{f}(\bar{x}) = \int_{T^n} f(\bar{t})e^{-2\pi i\bar{t}\cdot\bar{x}}d\bar{t}$$

$$\lim_{||\bar{x}||\to\infty} \hat{f}(\bar{x}) = 0$$

Fourier series on the n-cube

- Fourier analysis of this nature generalises near seamlessly to higher dimensions.
- Extend the definition of signals to real valued functions on the n-cube; $T^n = [0,1]^n$.

$$f(\bar{t}) \sim \sum_{\bar{x}\in\mathbb{N}^n} \hat{f}(\bar{x})e^{2\pi i\bar{t}}, \;\; \bar{x} = (x_1, x_2, ..., x_n)$$

$$\hat{f}(\bar{x}) = \int_{T^n} f(\bar{t})e^{-2\pi i\bar{t}\cdot\bar{x}}d\bar{t}$$

$$\lim_{||\bar{x}||\to\infty} \hat{f}(\bar{x}) = 0$$

Fourier series on the n-cube

- Fourier analysis of this nature generalises near seamlessly to higher dimensions.
- Extend the definition of signals to real valued functions on the n-cube; $T^n = [0,1]^n$.

$$f(\bar{t}) \sim \sum_{\bar{x} \in \mathbb{N}^n} \hat{f}(\bar{x}) e^{2\pi i \bar{t}}, \ \bar{x} = (x_1, x_2, ..., x_n)$$

$$\hat{f}(\bar{x}) = \int_{T^n} f(\bar{t}) e^{-2\pi i \bar{t} \cdot \bar{x}} d\bar{t}$$

$$\lim_{||\bar{x}|| \to \infty} \hat{f}(\bar{x}) = 0$$

Fourier series on the n-cube

- Fourier analysis of this nature generalises near seamlessly to higher dimensions.
- Extend the definition of signals to real valued functions on the n-cube; $T^n = [0,1]^n$.

$$f(\bar{t}) \sim \sum_{\bar{x} \in \mathbb{N}^n} \hat{f}(\bar{x}) e^{2\pi i \bar{t}}, \ \bar{x} = (x_1, x_2, ..., x_n)$$

$$\hat{f}(\bar{x}) = \int_{T^n} f(\bar{t}) e^{-2\pi i \bar{t} \cdot \bar{x}} d\bar{t}$$

$$\lim_{||\bar{x}|| \to \infty} \hat{f}(\bar{x}) = 0$$

Fourier series on the n-cube

- Fourier analysis of this nature generalises near seamlessly to higher dimensions.
- Extend the definition of signals to real valued functions on the n-cube; $T^n = [0,1]^n$.

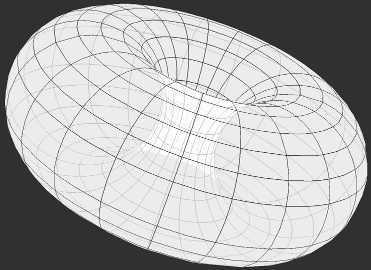
$$f(\bar{t}) \sim \sum_{\bar{x} \in \mathbb{N}^n} \hat{f}(\bar{x}) e^{2\pi i \bar{t}}, \ \bar{x} = (x_1, x_2, ..., x_n)$$

$$\hat{f}(\bar{x}) = \int_{T^n} f(\bar{t}) e^{-2\pi i \bar{t} \cdot \bar{x}} d\bar{t}$$

$$\lim_{||\bar{x}|| \to \infty} \hat{f}(\bar{x}) = 0$$

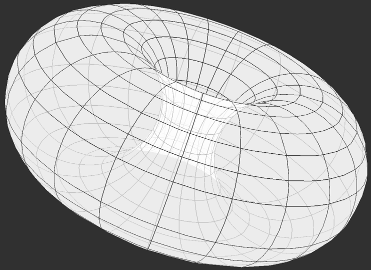
Periodicity

- Observe that the Fourier Series as defined on signals with domain T^n extend them to the entirety of $\mathbb{R}^n$.
- In particular, if we identify the endpoints of our intervals, our space is homeomorphic to the n-torus.



Periodicity

- Observe that the Fourier Series as defined on signals with domain T^n extend them to the entirety of $\mathbb{R}^n$.
- In particular, if we identify the endpoints of our intervals, our space is homeomorphic to the n-torus.



Results

- The rate of convergence of Fourier Series is governed by the smoothness of the signal.

$$f \in C^k \implies \hat{f}(n) \in O\left(\frac{1}{n^{k+1}}\right)$$

- For piece-wise continuous signals f with left and right first order derivatives everywhere on T;
- The Fourier series of f converges pointwise to f.
- The Cesaro means of the Fourier series converge pointwise to f.
- If f is continuous on T then the Fourier series of f converges to f uniformly with respect to $t \in T$.
- There exists a sequence of polynomials which converge to f uniformly with respect to $t \in T$

Results

- The rate of convergence of Fourier Series is governed by the smoothness of the signal.

$$f \in C^k \implies \hat{f}(n) \in O\left(\frac{1}{n^{k+1}}\right)$$

- For piece-wise continuous signals f with left and right first order derivatives everywhere on T;
- The Fourier series of f converges pointwise to f.
- The Cesaro means of the Fourier series converge pointwise to f.
- If f is continuous on T then the Fourier series of f converges to f uniformly with respect to $t \in T$.
- There exists a sequence of polynomials which converge to f uniformly with respect to $t \in T$

Results

- The rate of convergence of Fourier Series is governed by the smoothness of the signal.

$$f \in C^k \implies \hat{f}(n) \in O\left(\frac{1}{n^{k+1}}\right)$$

- For piece-wise continuous signals f with left and right first order derivatives everywhere on T;
- The Fourier series of f converges pointwise to f.
- The Cesaro means of the Fourier series converge pointwise to f.
- If f is continuous on T then the Fourier series of f converges to f uniformly with respect to $t \in T$.
- There exists a sequence of polynomials which converge to f uniformly with respect to $t \in T$

Results

- The rate of convergence of Fourier Series is governed by the smoothness of the signal.

$$f \in C^k \implies \hat{f}(n) \in O\left(\frac{1}{n^{k+1}}\right)$$

- For piece-wise continuous signals f with left and right first order derivatives everywhere on T;
- The Fourier series of f converges pointwise to f.
- The Cesaro means of the Fourier series converge pointwise to f.
- If f is continuous on T then the Fourier series of f converges to f uniformly with respect to $t \in T$.
- There exists a sequence of polynomials which converge to f uniformly with respect to $t \in T$

Results

- The rate of convergence of Fourier Series is governed by the smoothness of the signal.

$$f \in C^k \implies \hat{f}(n) \in O\left(\frac{1}{n^{k+1}}\right)$$

- For piece-wise continuous signals f with left and right first order derivatives everywhere on T;
- The Fourier series of f converges pointwise to f.
- The Cesaro means of the Fourier series converge pointwise to f.
- If f is continuous on T then the Fourier series of f converges to f uniformly with respect to $t \in T$.
- There exists a sequence of polynomials which converge to f uniformly with respect to $t \in T$

Results

- The rate of convergence of Fourier Series is governed by the smoothness of the signal.

$$f \in C^k \implies \hat{f}(n) \in O\left(\frac{1}{n^{k+1}}\right)$$

- For piece-wise continuous signals f with left and right first order derivatives everywhere on T;
- The Fourier series of f converges pointwise to f.
- The Cesaro means of the Fourier series converge pointwise to f.
- If f is continuous on T then the Fourier series of f converges to f uniformly with respect to $t \in T$.
- There exists a sequence of polynomials which converge to f uniformly with respect to $t \in T$

Results

- The rate of convergence of Fourier Series is governed by the smoothness of the signal.

$$f \in C^k \implies \hat{f}(n) \in O\left(\frac{1}{n^{k+1}}\right)$$

- For piece-wise continuous signals f with left and right first order derivatives everywhere on T;
- The Fourier series of f converges pointwise to f.
- The Cesaro means of the Fourier series converge pointwise to f.
- If f is continuous on T then the Fourier series of f converges to f uniformly with respect to $t \in T$.
- There exists a sequence of polynomials which converge to f uniformly with respect to $t \in T$

Methodology

- Integration techniques E.g. integration by parts, exploiting periodicity.
- Convolution, $f * g = \int_0^1 f(t)g(1-t)dt$.
- In particular, convolution with the Dirichlet and Fejer Kernels.

$$D_N(x) = \sum_{n=-N}^{N} e^{2\pi int} = \frac{\sin\left(2\pi(N+\frac{1}{2})x\right)}{\sin(\pi x)}$$

$$F_N(x) = \frac{1}{N}\sum_{n=0}^{N-1} D_n(x) = \frac{1}{N}\left(\frac{1-\cos(2\pi Nx)}{1-\cos(2\pi x)}\right)$$

- Epsilon-Delta proofs.

Methodology

- Integration techniques E.g. integration by parts, exploiting periodicity.
- Convolution, $f * g = \int_0^1 f(t)g(1-t)dt$.
- In particular, convolution with the Dirichlet and Fejer Kernels.

$$D_N(x) = \sum_{n=-N}^{N} e^{2\pi int} = \frac{\sin\left(2\pi(N+\frac{1}{2})x\right)}{\sin(\pi x)}$$

$$F_N(x) = \frac{1}{N}\sum_{n=0}^{N-1} D_n(x) = \frac{1}{N}\left(\frac{1-\cos(2\pi Nx)}{1-\cos(2\pi x)}\right)$$

- Epsilon-Delta proofs.

Methodology

- Integration techniques E.g. integration by parts, exploiting periodicity.
- Convolution, $f * g = \int_0^1 f(t)g(1-t)dt$.
- In particular, convolution with the Dirichlet and Fejer Kernels.

$$D_N(x) = \sum_{n=-N}^{N} e^{2\pi i n t} = \frac{\sin\left(2\pi(N+\frac{1}{2})x\right)}{\sin(\pi x)}$$

$$F_N(x) = \frac{1}{N}\sum_{n=0}^{N-1} D_n(x) = \frac{1}{N}\left(\frac{1-\cos(2\pi N x)}{1-\cos(2\pi x)}\right)$$

- Epsilon-Delta proofs.

Methodology

- Integration techniques E.g. integration by parts, exploiting periodicity.
- Convolution, $f * g = \int_0^1 f(t)g(1-t)dt$.
- In particular, convolution with the Dirichlet and Fejer Kernels.

$$D_N(x) = \sum_{n=-N}^{N} e^{2\pi int} = \frac{\sin\left(2\pi(N+\frac{1}{2})x\right)}{\sin(\pi x)}$$

$$F_N(x) = \frac{1}{N}\sum_{n=0}^{N-1} D_n(x) = \frac{1}{N}\left(\frac{1-\cos(2\pi Nx)}{1-\cos(2\pi x)}\right)$$

- Epsilon-Delta proofs.

Methodology

- Integration techniques E.g. integration by parts, exploiting periodicity.
- Convolution, $f * g = \int_0^1 f(t)g(1-t)dt$.
- In particular, convolution with the Dirichlet and Fejer Kernels.

$$D_N(x) = \sum_{n=-N}^{N} e^{2\pi int} = \frac{\sin\left(2\pi(N+\frac{1}{2})x\right)}{\sin(\pi x)}$$

$$F_N(x) = \frac{1}{N}\sum_{n=0}^{N-1} D_n(x) = \frac{1}{N}\left(\frac{1-\cos(2\pi Nx)}{1-\cos(2\pi x)}\right)$$

- Epsilon-Delta proofs.

Methodology

- Integration techniques E.g. integration by parts, exploiting periodicity.
- Convolution, $f * g = \int_0^1 f(t)g(1-t)dt$.
- In particular, convolution with the Dirichlet and Fejer Kernels.

$$D_N(x) = \sum_{n=-N}^{N} e^{2\pi int} = \frac{\sin\left(2\pi(N+\frac{1}{2})x\right)}{\sin(\pi x)}$$

$$F_N(x) = \frac{1}{N}\sum_{n=0}^{N-1} D_n(x) = \frac{1}{N}\left(\frac{1-\cos(2\pi Nx)}{1-\cos(2\pi x)}\right)$$

- Epsilon-Delta proofs.

The Fourier Transform

Motivating the Fourier Transform

- Information is often presented and interpreted in a naturally ordered manner;
- Sound pressure is temporal.
- Images have spatial composition.
- It's difficult to deduce the qualitative effect of the various components of some data set on human perception with analysis over the natural domain.
- Trends in the frequency domain are much more reliably and accurately correlated with perception.

Motivating the Fourier Transform

- Information is often presented and interpreted in a naturally ordered manner;
- Sound pressure is temporal.
- Images have spatial composition.
- It's difficult to deduce the qualitative effect of the various components of some data set on human perception with analysis over the natural domain.
- Trends in the frequency domain are much more reliably and accurately correlated with perception.

Motivating the Fourier Transform

- Information is often presented and interpreted in a naturally ordered manner;
- Sound pressure is temporal.
- Images have spatial composition.
- It's difficult to deduce the qualitative effect of the various components of some data set on human perception with analysis over the natural domain.
- Trends in the frequency domain are much more reliably and accurately correlated with perception.

Motivating the Fourier Transform

- Information is often presented and interpreted in a naturally ordered manner;
- Sound pressure is temporal.
- Images have spatial composition.
- It's difficult to deduce the qualitative effect of the various components of some data set on human perception with analysis over the natural domain.
- Trends in the frequency domain are much more reliably and accurately correlated with perception.

Motivating the Fourier Transform

- Information is often presented and interpreted in a naturally ordered manner;
- Sound pressure is temporal.
- Images have spatial composition.
- It's difficult to deduce the qualitative effect of the various components of some data set on human perception with analysis over the natural domain.
- Trends in the frequency domain are much more reliably and accurately correlated with perception.

The Fourier Transform

- We may consider real valued integrable functions on $\mathbb{R}$ signals with arbitrarily large period.
- In turn, the 'distance' between frequencies becomes arbitrarily small.
- In particular, the function $\hat{f}(n)$ on the integers is extended to a continuous function over the reals.

$$\mathcal{F}f(s) = \int_{-\infty}^{\infty} f(t)e^{-2\pi i s t}dt$$

The Fourier Transform

- We may consider real valued integrable functions on $\mathbb{R}$ signals with arbitrarily large period.
- In turn, the 'distance' between frequencies becomes arbitrarily small.
- In particular, the function $\hat{f}(n)$ on the integers is extended to a continuous function over the reals.

$$\mathcal{F}f(s) = \int_{-\infty}^{\infty} f(t)e^{-2\pi i s t}dt$$

The Fourier Transform

- We may consider real valued integrable functions on $\mathbb{R}$ signals with arbitrarily large period.
- In turn, the 'distance' between frequencies becomes arbitrarily small.
- In particular, the function $\hat{f}(n)$ on the integers is extended to a continuous function over the reals.

$$\mathcal{F}f(s) = \int_{-\infty}^{\infty} f(t)e^{-2\pi ist}dt$$

The Fourier Transform

- We may consider real valued integrable functions on $\mathbb{R}$ signals with arbitrarily large period.
- In turn, the 'distance' between frequencies becomes arbitrarily small.
- In particular, the function $\hat{f}(n)$ on the integers is extended to a continuous function over the reals.

$$\mathcal{F}f(s) = \int_{-\infty}^{\infty} f(t)e^{-2\pi i s t}dt$$

The Fourier Transform on $\mathbb{R}^n$

- Data is often multidimensional;
- time
- space
- colour
- We can extend the fourier transform to higher dimensions analogously to the Fourier Series.

$$\mathcal{F}f(\bar{s}) = \int_{\mathbb{R}^n} f(\bar{t})\, e^{-2\pi i \bar{t}\cdot\bar{s}} d\bar{t}$$

The Fourier Transform on $\mathbb{R}^n$

- Data is often multidimensional;
- time
- space
- colour
- We can extend the fourier transform to higher dimensions analogously to the Fourier Series.

$$\mathcal{F}f(\bar{s}) = \int_{\mathbb{R}^n} f(\bar{t})\, e^{-2\pi i \bar{t}\cdot\bar{s}} d\bar{t}$$

The Fourier Transform on $\mathbb{R}^n$

- Data is often multidimensional;
- time
- space
- colour
- We can extend the fourier transform to higher dimensions analogously to the Fourier Series.

$$\mathcal{F}f(\bar{s}) = \int_{\mathbb{R}^n} f(\bar{t})\, e^{-2\pi i \bar{t}\cdot\bar{s}} d\bar{t}$$

The Fourier Transform on $\mathbb{R}^n$

- Data is often multidimensional;
- time
- space
- colour
- We can extend the fourier transform to higher dimensions analogously to the Fourier Series.

$$\mathcal{F}f(\bar{s}) = \int_{\mathbb{R}^n} f(\bar{t})\, e^{-2\pi i \bar{t}\cdot\bar{s}} d\bar{t}$$

The Fourier Transform on $\mathbb{R}^n$

- Data is often multidimensional;
- time
- space
- colour
- We can extend the fourier transform to higher dimensions analogously to the Fourier Series.

$$\mathcal{F}f(\bar{s}) = \int_{\mathbb{R}^n} f(\bar{t})\, e^{-2\pi i \bar{t}\cdot\bar{s}} d\bar{t}$$

The Fourier Transform on $\mathbb{R}^n$

- Data is often multidimensional;
  - time
  - space
  - colour
- We can extend the fourier transform to higher dimensions analogously to the Fourier Series.

$$\mathcal{F}f(\bar{s}) = \int_{\mathbb{R}^n} f(\bar{t})\, e^{-2\pi i \bar{t}\cdot\bar{s}} d\bar{t}$$

Discrete information

- Signals realised in the world at large are discrete, not continuous.
- Implementations of the Fourier transform need to be designed appropriately.
- In particular, a discrete analogue is needed.
- "It Works to the extent that a large fraction of the world's economy depends on it working, and that's not a bad measure of success. Consider this a proof by economics."
- Brad G Osgood - Lectures on the Fourier Transform and Its Applications

Discrete information

- Signals realised in the world at large are discrete, not continuous.
- Implementations of the Fourier transform need to be designed appropriately.
- In particular, a discrete analogue is needed.
- "It Works to the extent that a large fraction of the world's economy depends on it working, and that's not a bad measure of success. Consider this a proof by economics."
- Brad G Osgood - Lectures on the Fourier Transform and Its Applications

Discrete information

- Signals realised in the world at large are discrete, not continuous.
- Implementations of the Fourier transform need to be designed appropriately.
- In particular, a discrete analogue is needed.
- "It Works to the extent that a large fraction of the world's economy depends on it working, and that's not a bad measure of success. Consider this a proof by economics."
- Brad G Osgood - Lectures on the Fourier Transform and Its Applications

Discrete information

- Signals realised in the world at large are discrete, not continuous.
- Implementations of the Fourier transform need to be designed appropriately.
- In particular, a discrete analogue is needed.
- "It Works to the extent that a large fraction of the world's economy depends on it working, and that's not a bad measure of success. Consider this a proof by economics."
- Brad G Osgood - Lectures on the Fourier Transform and Its Applications

Discrete information

- Signals realised in the world at large are discrete, not continuous.
- Implementations of the Fourier transform need to be designed appropriately.
- In particular, a discrete analogue is needed.
- "It Works to the extent that a large fraction of the world's economy depends on it working, and that's not a bad measure of success. Consider this a proof by economics."
- Brad G Osgood - Lectures on the Fourier Transform and Its Applications

DFT

- The Discrete Fourier Transform (DFT) is discretised in both time and frequency.
- We may choose two of three parameters, the sampling rate, the length of the interval and the total number of samples
- Judicious choices of these (along with savvy implementation) are key to maximising the efficacy of the DFT.

DFT

- The Discrete Fourier Transform (DFT) is discretised in both time and frequency.
- We may choose two of three parameters, the sampling rate, the length of the interval and the total number of samples
- Judicious choices of these (along with savvy implementation) are key to maximising the efficacy of the DFT.

DFT

- The Discrete Fourier Transform (DFT) is discretised in both time and frequency.
- We may choose two of three parameters, the sampling rate, the length of the interval and the total number of samples
- Judicious choices of these (along with savvy implementation) are key to maximising the efficacy of the DFT.

DFT

- Choose N to be the number of samples (and the number of frequencies).
- f is now defined on a discrete set of times with cardinality N

$$\omega_N := e^{\frac{2\pi i}{N}}$$

$$\underline{\mathcal{F}}\,\underline{f}[s] = \sum_{k=0}^{N-1} \underline{f}[t_k]\omega_N^{-ks}$$

DFT

- Choose N to be the number of samples (and the number of frequencies).
- f is now defined on a discrete set of times with cardinality N

$$\omega_N := e^{\frac{2\pi i}{N}}$$

$$\underline{\mathcal{F}}\underline{f}[s] = \sum_{k=0}^{N-1} \underline{f}[t_k]\omega_N^{-ks}$$

DFT

- Choose N to be the number of samples (and the number of frequencies).
- f is now defined on a discrete set of times with cardinality N

$$\omega_N := e^{\frac{2\pi i}{N}}$$

$$\underline{\mathcal{F}}\underline{f}[s] = \sum_{k=0}^{N-1} \underline{f}[t_k]\omega_N^{-ks}$$

DFT

- Choose N to be the number of samples (and the number of frequencies).
- $\underline{f}$ is now defined on a discrete set of times with cardinality N

$$\omega_N := e^{\frac{2\pi i}{N}}$$

$$\underline{\mathcal{F}}\,\underline{f}[s] = \sum_{k=0}^{N-1} \underline{f}[t_k]\omega_N^{-ks}$$

DFT Matrix

- The DFT as defined lends itself naturally to matrix form.

$$\begin{bmatrix} \underline{\mathcal{F}}\,\underline{f}[s_0] \\ \underline{\mathcal{F}}\,\underline{f}[s_1] \\ \vdots \\ \underline{\mathcal{F}}\,\underline{f}[s_{N-2}] \\ \underline{\mathcal{F}}\,\underline{f}[s_{N-1}] \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_N^{-1\cdot 1} & \omega_N^{-1\cdot 2} & \cdots & \omega_N^{-1\cdot(N-1)} \\ 1 & \omega_N^{-2\cdot 1} & \omega_N^{-2\cdot 2} & \cdots & \omega_N^{-2\cdot(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_N^{-(N-1)\cdot 1} & \omega_N^{-(N-1)\cdot 2} & \cdots & \omega_N^{(N-1)^2} \end{bmatrix} \begin{bmatrix} \underline{f}[t_0] \\ \underline{f}[t_1] \\ \vdots \\ \underline{f}[t_{N-2}] \\ \underline{f}[t_{N-1}] \end{bmatrix}$$

- A naive implementation of the DFT then has complexity $O(N^2)$

DFT Matrix

- The DFT as defined lends itself naturally to matrix form.

$$\begin{bmatrix} \underline{\mathcal{F}}\underline{f}[s_0] \\ \underline{\mathcal{F}}\underline{f}[s_1] \\ \vdots \\ \underline{\mathcal{F}}\underline{f}[s_{N-2}] \\ \underline{\mathcal{F}}\underline{f}[s_{N-1}] \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_N^{-1\cdot 1} & \omega_N^{-1\cdot 2} & \cdots & \omega_N^{-1\cdot(N-1)} \\ 1 & \omega_N^{-2\cdot 1} & \omega_N^{-2\cdot 2} & \cdots & \omega_N^{-2\cdot(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_N^{-(N-1)\cdot 1} & \omega_N^{-(N-1)\cdot 2} & \cdots & \omega_N^{(N-1)^2} \end{bmatrix} \begin{bmatrix} \underline{f}[t_0] \\ \underline{f}[t_1] \\ \vdots \\ \underline{f}[t_{N-2}] \\ \underline{f}[t_{N-1}] \end{bmatrix}$$

- A naive implementation of the DFT then has complexity $O(N^2)$

DFT Matrix

- The DFT as defined lends itself naturally to matrix form.

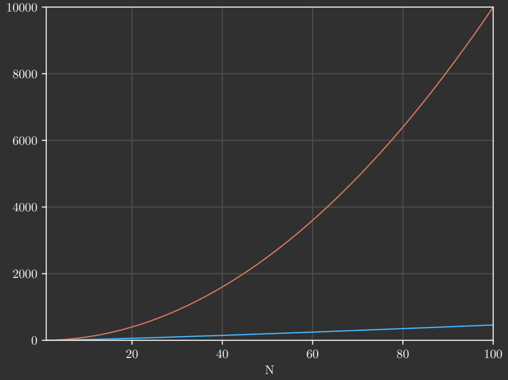
$$\begin{bmatrix} \underline{\mathcal{F}}\underline{f}[s_0] \\ \underline{\mathcal{F}}\underline{f}[s_1] \\ \vdots \\ \underline{\mathcal{F}}\underline{f}[s_{N-2}] \\ \underline{\mathcal{F}}\underline{f}[s_{N-1}] \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_N^{-1\cdot 1} & \omega_N^{-1\cdot 2} & \cdots & \omega_N^{-1\cdot(N-1)} \\ 1 & \omega_N^{-2\cdot 1} & \omega_N^{-2\cdot 2} & \cdots & \omega_N^{-2\cdot(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_N^{-(N-1)\cdot 1} & \omega_N^{-(N-1)\cdot 2} & \cdots & \omega_N^{(N-1)^2} \end{bmatrix} \begin{bmatrix} \underline{f}[t_0] \\ \underline{f}[t_1] \\ \vdots \\ \underline{f}[t_{N-2}] \\ \underline{f}[t_{N-1}] \end{bmatrix}$$

- A naive implementation of the DFT then has complexity $O(N^2)$

FFT

- Fast Fourier Transforms (FFTs) are a class of algorithms which implement the Discrete Fourier Transform.
- Typically, these are divide and conquer algorithms which achieve $O(N \log N)$ complexity through the use of recursion.
- "The most important numerical algorithm of our lifetime"
  - Gilbert Strang

- N^2 in orange.
- $N \log N$ in blue.



Applications

Huffman Encoding

- Huffman Encoding is a scheme to save memory by exploiting the frequency of symbols.
- A tree is constructed based on the frequency with which symbols occur and they are then assigned a binary encoding based on this tree.
- Symbols that ocur more frequently are assigned shorter encodings.
- If one is willing to replace a significant portion of a string with a single symbol, the space required to store information can be reduced considerably.
- This (amongst other entropy encoding techniques) is the basis of all lossy Fourier Transformation compression Algorithms.
- The key is to determine what information can be discarded with little effect.

Huffman Encoding

- Huffman Encoding is a scheme to save memory by exploiting the frequency of symbols.
- A tree is constructed based on the frequency with which symbols occur and they are then assigned a binary encoding based on this tree.
- Symbols that ocur more frequently are assigned shorter encodings.
- If one is willing to replace a significant portion of a string with a single symbol, the space required to store information can be reduced considerably.
- This (amongst other entropy encoding techniques) is the basis of all lossy Fourier Transformation compression Algorithms.
- The key is to determine what information can be discarded with little effect.

Huffman Encoding

- Huffman Encoding is a scheme to save memory by exploiting the frequency of symbols.
- A tree is constructed based on the frequency with which symbols occur and they are then assigned a binary encoding based on this tree.
- Symbols that ocur more frequently are assigned shorter encodings.
- If one is willing to replace a significant portion of a string with a single symbol, the space required to store information can be reduced considerably.
- This (amongst other entropy encoding techniques) is the basis of all lossy Fourier Transformation compression Algorithms.
- The key is to determine what information can be discarded with little effect.

Huffman Encoding

- Huffman Encoding is a scheme to save memory by exploiting the frequency of symbols.
- A tree is constructed based on the frequency with which symbols occur and they are then assigned a binary encoding based on this tree.
- Symbols that ocur more frequently are assigned shorter encodings.
- If one is willing to replace a significant portion of a string with a single symbol, the space required to store information can be reduced considerably.
- This (amongst other entropy encoding techniques) is the basis of all lossy Fourier Transformation compression Algorithms.
- The key is to determine what information can be discarded with little effect.

Huffman Encoding

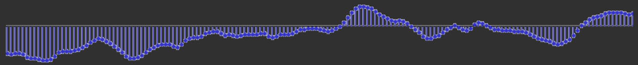
- Huffman Encoding is a scheme to save memory by exploiting the frequency of symbols.
- A tree is constructed based on the frequency with which symbols occur and they are then assigned a binary encoding based on this tree.
- Symbols that ocur more frequently are assigned shorter encodings.
- If one is willing to replace a significant portion of a string with a single symbol, the space required to store information can be reduced considerably.
- This (amongst other entropy encoding techniques) is the basis of all lossy Fourier Transformation compression Algorithms.
- The key is to determine what information can be discarded with little effect.

Huffman Encoding

- Huffman Encoding is a scheme to save memory by exploiting the frequency of symbols.
- A tree is constructed based on the frequency with which symbols occur and they are then assigned a binary encoding based on this tree.
- Symbols that ocur more frequently are assigned shorter encodings.
- If one is willing to replace a significant portion of a string with a single symbol, the space required to store information can be reduced considerably.
- This (amongst other entropy encoding techniques) is the basis of all lossy Fourier Transformation compression Algorithms.
- The key is to determine what information can be discarded with little effect.

Audio

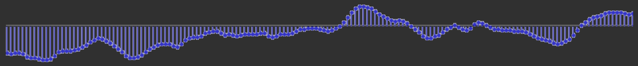
- Audio is a natural candidate for the Fourier transform in light of it's well known association with frequency.
- The Fourier transform of an audio signal conveys the amplitude of each frequency present.



- The Ecstasy of Gold - Ennio Morricone

Audio

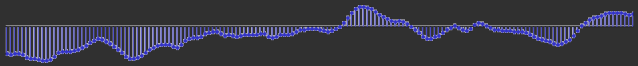
- Audio is a natural candidate for the Fourier transform in light of it's well known association with frequency.
- The Fourier transform of an audio signal conveys the amplitude of each frequency present.



- The Ecstasy of Gold - Ennio Morricone

Audio

- Audio is a natural candidate for the Fourier transform in light of it's well known association with frequency.
- The Fourier transform of an audio signal conveys the amplitude of each frequency present.



- The Ecstasy of Gold - Ennio Morricone

Psychoacoustics

- Humans only perceive a narrow band of frequencies, $\sim$ 20Hz - 20kHz
- Moreover, within this band of frequencies our capacity to process the volume of data available to us is severly limited.
- Current (experimentally determined) models posit the existence of 24 'critical bands', frequency ranges within which one can only discern a single dominant frequency in some temporal neighbourhood.
- By partitioning the signal into many short time frames, the Fourier transform can be used to determine the dominant tone of each and discard the remainder to save space via Huffman Encoding.

Psychoacoustics

- Humans only perceive a narrow band of frequencies, $\sim$ 20Hz - 20kHz
- Moreover, within this band of frequencies our capacity to process the volume of data available to us is severly limited.
- Current (experimentally determined) models posit the existence of 24 'critical bands', frequency ranges within which one can only discern a single dominant frequency in some temporal neighbourhood.
- By partitioning the signal into many short time frames, the Fourier transform can be used to determine the dominant tone of each and discard the remainder to save space via Huffman Encoding.

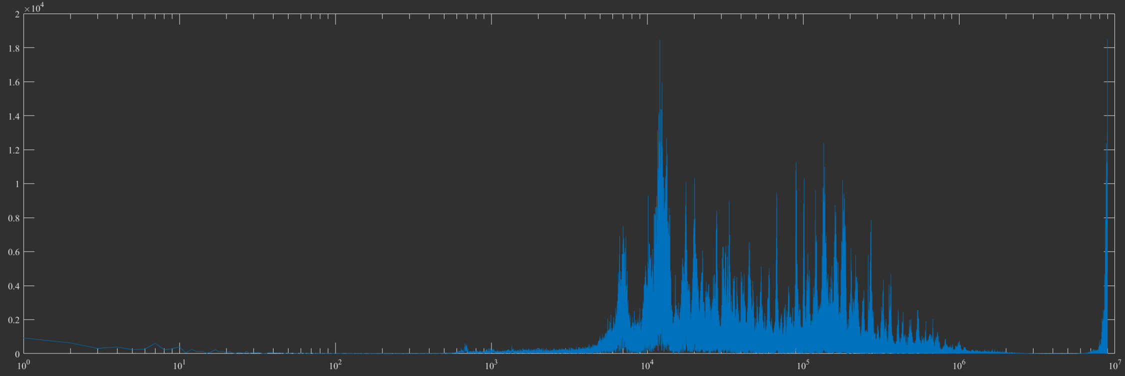
Psychoacoustics

- Humans only perceive a narrow band of frequencies, $\sim$ 20Hz - 20kHz
- Moreover, within this band of frequencies our capacity to process the volume of data available to us is severly limited.
- Current (experimentally determined) models posit the existence of 24 'critical bands', frequency ranges within which one can only discern a single dominant frequency in some temporal neighbourhood.
- By partitioning the signal into many short time frames, the Fourier transform can be used to determine the dominant tone of each and discard the remainder to save space via Huffman Encoding.

Psychoacoustics

- Humans only perceive a narrow band of frequencies, $\sim$ 20Hz - 20kHz
- Moreover, within this band of frequencies our capacity to process the volume of data available to us is severly limited.
- Current (experimentally determined) models posit the existence of 24 'critical bands', frequency ranges within which one can only discern a single dominant frequency in some temporal neighbourhood.
- By partitioning the signal into many short time frames, the Fourier transform can be used to determine the dominant tone of each and discard the remainder to save space via Huffman Encoding.

The Ecstasy of Gold FFT



JPEG Compression

- Compression of images operates in much the same way.
- The image is interpreted as a 2D array of tuples, typically representing information about hue and brightness.
- The array is divided into smaller arrays with dimensions $\sim 2^6 \times 2^6$.
- A 2 dimensional FFT is performed on each of the sub-arrays, and certain frequencies are discarded according to some experimentally determined and refined scheme.
- In particular the result is multiplied by a proprietary quantisation matrix.
- Entries of the proudct less than 1 are deleted, and the Fourier transformed matrix is multiplied by the inverse of the quantisation matrix before being Fourier inverted.

JPEG Compression

- Compression of images operates in much the same way.
- The image is interpreted as a 2D array of tuples, typically representing information about hue and brightness.
- The array is divided into smaller arrays with dimensions $\sim 2^6 \times 2^6$.
- A 2 dimensional FFT is performed on each of the sub-arrays, and certain frequencies are discarded according to some experimentally determined and refined scheme.
- In particular the result is multiplied by a proprietary quantisation matrix.
- Entries of the proudct less than 1 are deleted, and the Fourier transformed matrix is multiplied by the inverse of the quantisation matrix before being Fourier inverted.

JPEG Compression

- Compression of images operates in much the same way.
- The image is interpreted as a 2D array of tuples, typically representing information about hue and brightness.
- The array is divided into smaller arrays with dimensions $\sim 2^6 \times 2^6$.
- A 2 dimensional FFT is performed on each of the sub-arrays, and certain frequencies are discarded according to some experimentally determined and refined scheme.
- In particular the result is multiplied by a proprietary quantisation matrix.
- Entries of the proudct less than 1 are deleted, and the Fourier transformed matrix is multiplied by the inverse of the quantisation matrix before being Fourier inverted.

JPEG Compression

- Compression of images operates in much the same way.
- The image is interpreted as a 2D array of tuples, typically representing information about hue and brightness.
- The array is divided into smaller arrays with dimensions $\sim 2^6 \times 2^6$.
- A 2 dimensional FFT is performed on each of the sub-arrays, and certain frequencies are discarded according to some experimentally determined and refined scheme.
- In particular the result is multiplied by a proprietary quantisation matrix.
- Entries of the proudct less than 1 are deleted, and the Fourier transformed matrix is multiplied by the inverse of the quantisation matrix before being Fourier inverted.

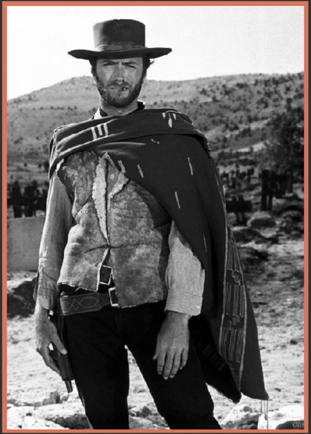
JPEG Compression

- Compression of images operates in much the same way.
- The image is interpreted as a 2D array of tuples, typically representing information about hue and brightness.
- The array is divided into smaller arrays with dimensions $\sim 2^6 \times 2^6$.
- A 2 dimensional FFT is performed on each of the sub-arrays, and certain frequencies are discarded according to some experimentally determined and refined scheme.
- In particular the result is multiplied by a proprietary quantisation matrix.
- Entries of the proudct less than 1 are deleted, and the Fourier transformed matrix is multiplied by the inverse of the quantisation matrix before being Fourier inverted.

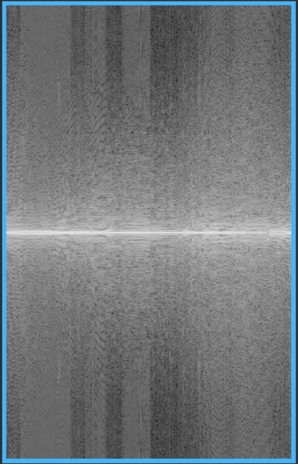
JPEG Compression

- Compression of images operates in much the same way.
- The image is interpreted as a 2D array of tuples, typically representing information about hue and brightness.
- The array is divided into smaller arrays with dimensions $\sim 2^6 \times 2^6$.
- A 2 dimensional FFT is performed on each of the sub-arrays, and certain frequencies are discarded according to some experimentally determined and refined scheme.
- In particular the result is multiplied by a proprietary quantisation matrix.
- Entries of the proudct less than 1 are deleted, and the Fourier transformed matrix is multiplied by the inverse of the quantisation matrix before being Fourier inverted.

Blondie



Blondie FFT



Lossy Compression

- Such compression algorithms are termed 'lossy'.
- This is because information originally encoded is lost in the process, and can not be recovered via Fourier inversion.
- Typically this is an acceptable trade off, observable discrepancies between the original and are minor and scarce.
- MP3 compression ratios vary depending on user selected parameters which govern the quality of the output, namely bit and sample rates.
- At 128 kilobits per second and 44.1kHz (the modal choice of parameters) compression ratios are on average 11 to 1.
- JPEG typically achieves a 10 to 1 compression ratio.

Lossy Compression

- Such compression algorithms are termed 'lossy'.
- This is because information originally encoded is lost in the process, and can not be recovered via Fourier inversion.
- Typically this is an acceptable trade off, observable discrepancies between the original and are minor and scarce.
- MP3 compression ratios vary depending on user selected parameters which govern the quality of the output, namely bit and sample rates.
- At 128 kilobits per second and 44.1kHz (the modal choice of parameters) compression ratios are on average 11 to 1.
- JPEG typically achieves a 10 to 1 compression ratio.

Lossy Compression

- Such compression algorithms are termed 'lossy'.
- This is because information originally encoded is lost in the process, and can not be recovered via Fourier inversion.
- Typically this is an acceptable trade off, observable discrepancies between the original and are minor and scarce.
- MP3 compression ratios vary depending on user selected parameters which govern the quality of the output, namely bit and sample rates.
- At 128 kilobits per second and 44.1kHz (the modal choice of parameters) compression ratios are on average 11 to 1.
- JPEG typically achieves a 10 to 1 compression ratio.

Lossy Compression

- Such compression algorithms are termed 'lossy'.
- This is because information originally encoded is lost in the process, and can not be recovered via Fourier inversion.
- Typically this is an acceptable trade off, observable discrepancies between the original and are minor and scarce.
- MP3 compression ratios vary depending on user selected parameters which govern the quality of the output, namely bit and sample rates.
- At 128 kilobits per second and 44.1kHz (the modal choice of parameters) compression ratios are on average 11 to 1.
- JPEG typically achieves a 10 to 1 compression ratio.

Lossy Compression

- Such compression algorithms are termed 'lossy'.
- This is because information originally encoded is lost in the process, and can not be recovered via Fourier inversion.
- Typically this is an acceptable trade off, observable discrepancies between the original and are minor and scarce.
- MP3 compression ratios vary depending on user selected parameters which govern the quality of the output, namely bit and sample rates.
- At 128 kilobits per second and 44.1kHz (the modal choice of parameters) compression ratios are on average 11 to 1.
- JPEG typically achieves a 10 to 1 compression ratio.

Lossy Compression

- Such compression algorithms are termed 'lossy'.
- This is because information originally encoded is lost in the process, and can not be recovered via Fourier inversion.
- Typically this is an acceptable trade off, observable discrepancies between the original and are minor and scarce.
- MP3 compression ratios vary depending on user selected parameters which govern the quality of the output, namely bit and sample rates.
- At 128 kilobits per second and 44.1kHz (the modal choice of parameters) compression ratios are on average 11 to 1.
- JPEG typically achieves a 10 to 1 compression ratio.

Acknowledgments

Special thanks to Prof. Daniel Daners for his guidance and support throughout the semester.

References

- Katznelson, Y. *An Introduction to Harmonic Analysis.* 1st ed. (John Wiley Sons, Inc, 1968).
- Osgood, B. G. *Lectures on the Fourier Transform and Its Applications.* (AMS, 2019).
- Raissi, R. *The Theory Behind Mp3.* http://www.mp3-tech.org/programmer/docs/mp3_theory.pdf.

Thank You For Your Time