Cosine Transform

Monday, February 6, 2017 10:28 AM

The cosine transform is an alternative to the Fourier series — it is an orthobasis for $[0,1]$

we can choose other intervals

1) The cosine transform bases and coefficients are real-valued (FS → complex)

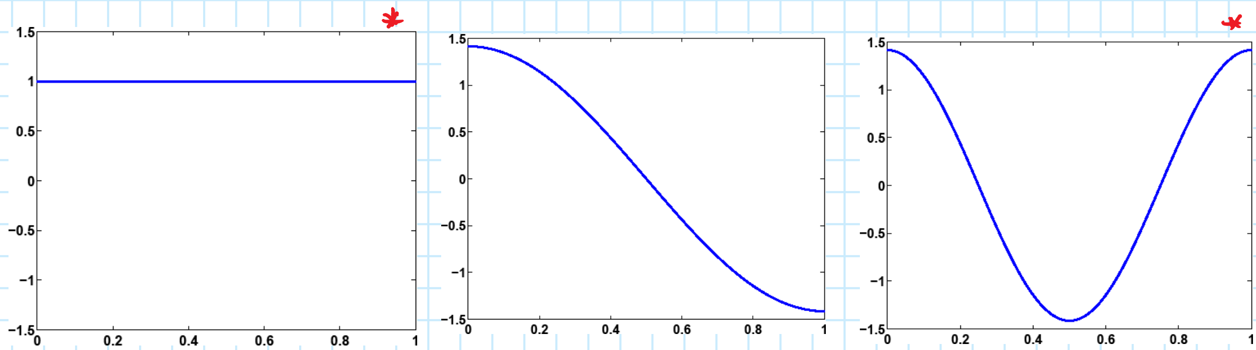
2) The basis functions have different symmetries

There are multiple different cosine transforms

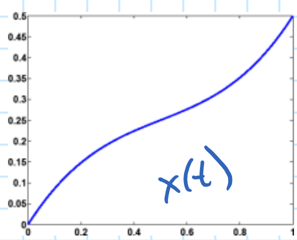
The discrete version of cosine-I (the DCT or DCT-I) is used in JPEG and MPEG compression.

The cosine-I basis functions for $t \in [0,1]$ are

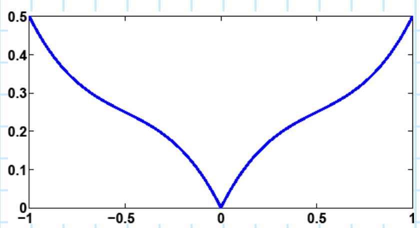
$$\psi_k(t) = \begin{cases} 1 & k=0 \\ \sqrt{2}\cos(\pi k t) & k=1,2,\ldots \end{cases}$$



Let $x(t)$ be a signal on the interval $[0,1]$. Let $\tilde{x}(t)$ be its "reflected extension" on $[-1,1]$



$$\tilde{x}(t) = \begin{cases} x(-t) & -1 \le t \le 0 \\ x(t) & 0 \le t \le 1 \end{cases}$$



$\tilde{x}(t)$

Cosine Transform cont.

Wednesday, February 8, 2017 9:31 AM

$$\tilde{x}(t) = \sum_{k=-\infty}^{\infty} \alpha_k e^{j\pi k t}$$

since $\tilde{x}(t)$ is real, we have $\alpha_{-k} = \bar{\alpha}_k$

$$= \alpha_0 + \sum_{k=1}^{\infty} \left(\bar{\alpha}_k e^{-j\pi k t} + \alpha_k e^{j\pi k t} \right)$$

$$\boxed{\tilde{x}(t) = a_0 + \sum_{k=1}^{\infty} a_k \cos(\pi k t + \phi_k)}$$

$$a_0 = \alpha_0, \quad a_k = 2\,\mathrm{Re}\{\alpha_k\} \quad \phi_k = ?$$

since $\tilde{x}(t)$ is an even function, There is no imaginary component of α_k so $\phi_k = 0$

This works for any symmetric $\tilde{x}(t)$ on $[-1, 1]$ — we can create this from any $x(t)$ on $[0, 1]$.

Therefore, we have a mapping from $x(t)$ on $[0, 1]$ to $\{a_k\}_{k=0}^{\infty}$

Cosine Transform cont.

Wednesday, February 8, 2017 9:43 AM

We can get our $\{a_k\}$'s for a given $x(t)$ on $[0,1]$ if we establish $\{\psi_k(t)\}_{k=0}^{\infty}$ as an orthobasis.

$$\langle \psi_k, \psi_\ell \rangle = 2\int_0^1 \cos(\pi k t)\cos(\pi \ell t)\,dt = \begin{cases} 1 & k=\ell \\ 0 & k\neq \ell \end{cases}$$

One way of thinking about the cosine transform is that it is like an oversampled Fourier series – we have frequencies spaced at multiples of π rather than 2π. Then we only take the real part.

Discrete Cosine Transform

Wednesday, February 8, 2017 9:50 AM

Fourier transforms

time-domain	frequency
discrete	periodic
continuous	non-periodic
periodic	discrete
non-periodic	continuous

Freq \ Time	discrete	cont.
disc.	DCT DFT T: periodic F: periodic	Cosine Trans. Fourier Series T: periodic
cont.	DTFT F: periodic	Fourier Transform

The DCT is analogous to the DFT. It is a sampled version of the cosine transform

The DCT basis functions

$$\psi_k[n] = \begin{cases} \frac{1}{\sqrt{N}} & k=0 \\ \sqrt{\frac{2}{N}} \cos\left(\frac{\pi k}{N}\left(n+\frac{1}{2}\right)\right) & k=1,\dots,N-1 \end{cases}$$

$$\sum_{n=0}^{N-1} \psi_k[n]\, \psi_\ell[n] = \begin{cases} 1 & k=\ell \\ 0 & k\neq \ell \end{cases}$$

notice, we use half sample points $(n+\frac{1}{2})$ instead of n in the expression above

it turns out that this works best for compression

We can compute the DCT using the FFT for fast computation.

2D - Cosine Transform

Definition: Let $\{\psi_k(t)\}_{k\geq 0}$ be the cosine-I basis.

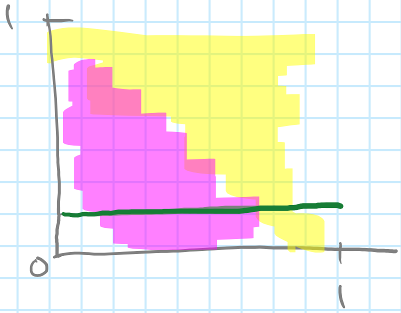
Set $\psi^{2D}_{k_1,k_2}(s,t) = \psi_{k_1}(s)\, \psi_{k_2}(t)$

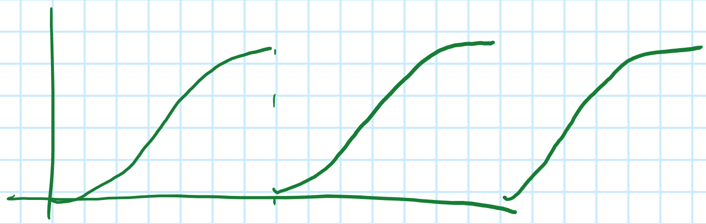
Then $\{\psi^{2D}_{k_1 k_2}(s,t)\}_{k_1,k_2\in\mathbb{N}}$ is an orthonormal basis for $L_2([0,1]^2)$.

This is true in general, that if $\{\phi_k(t)\}$ is an orthobasis on $L_2([0,1])$ then $\phi_{k_1,k_2}(s,t) = \phi_{k_1}(s)\, \phi_{k_2}(t)$ is an orthobasis on $L_2([0,1]^2)$

2-D Cosine Transform

Wednesday, February 8, 2017 10:16 AM





we need to show that

$$\int_0^1\int_0^1 \left(\phi_{k_1 k_2}(s,t)\right)\left(\phi_{l_1 l_2}(s,t)\right) ds\,dt$$

$$= \begin{cases} 1 & k_1 = l_1,\ k_2 = l_2 \\ 0 & \text{otherwise} \end{cases}$$

Then we need to show that $\{\phi_{k_1 k_2}(s,t)\}$ spans $[0,1]^2$ ... (part of HW 4)

outline of proof

step 1: show that the $\phi_{k_1,k_2}(s,t)$ are orthonormal.
← this part is straightforward

step 2: assuming that s is fixed at $s = s_0$
so then we know

$$X(s_0,t) = \sum_k \alpha_k \Psi_k(t)$$

function of our choice for s.
$\alpha_k(s)$

step 3: fix k and decompose the $\alpha_k(s)$'s

⋮

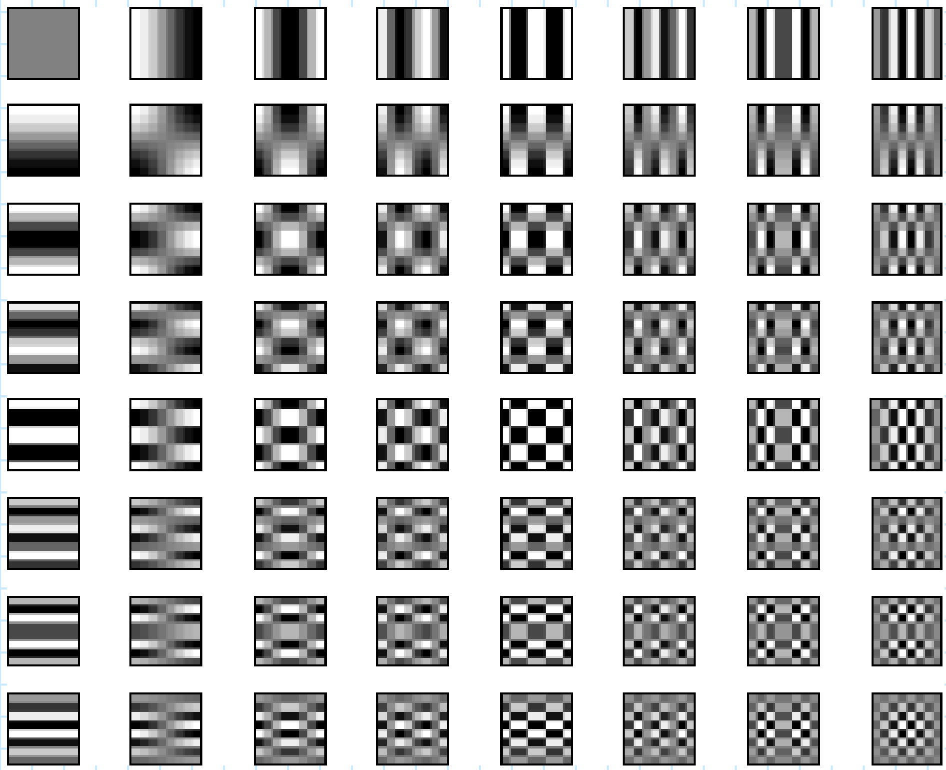
2-D DCT

Monday, February 13, 2017 8:55 AM

Definition: Let $\{\psi_k[n]\}_{0 \le k \le N-1}$ be the DCT basis as described previously.

Set $\psi^{2D}_{j,k}[m,n] = \psi_j[m]\,\psi_k[n]$

Then $\{\psi^{2D}_{j,k}[m,n]\}_{0 \le j,k \le N-1}$ is an orthonormal basis for $\mathbb{R}^N \times \mathbb{R}^N$.



$N=8$

used in JPEG and MPEG compression

The 2-D coefficients are indexed by two integers and so are naturally arranged on a grid as well:

$$
\begin{matrix}
\alpha_{0,0} & \alpha_{0,1} & \alpha_{0,2} & \cdots & \alpha_{0,N-1} \\
\alpha_{1,0} & \alpha_{1,1} & & & \vdots \\
\vdots & & \ddots & & \\
\alpha_{N-1,0} & & \cdots & & \alpha_{N-1,N-1}
\end{matrix}
$$

Image Compression

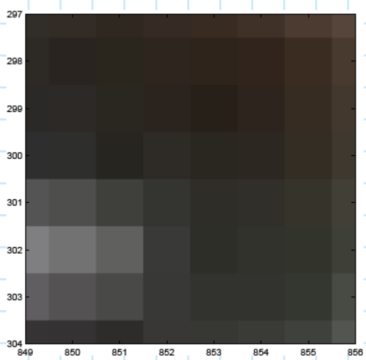
Monday, February 13, 2017 9:26 AM

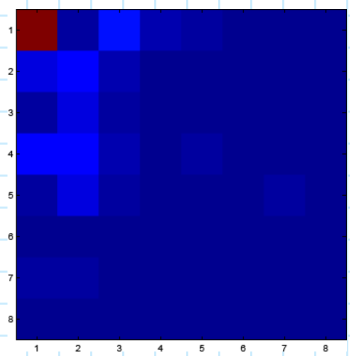
JPEG – it is built around the DCT

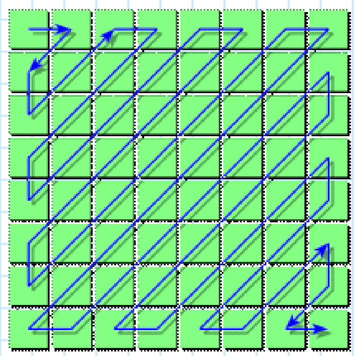
↳ for images, the DCT tends to concentrate energy in the low-frequency coefficients.

Basic operation

1) Divide the image into 8 × 8 blocks of pixels
2) Take the DCT of each block
3) Quantize the coefficients ~ the general effect is to keep large coefficients and discard small ones.
4) Losslessly compress/encode the result.







The coeffients are read in the zig-zag pattern shown

- JPEG does not just keep or kill coefficients, it quantizes them using a fixed mask

$$Q = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}.$$

$$\tilde{\alpha}_{j,k} = Q_{j,k} \cdot \text{round}\left(\frac{\alpha_{j,k}}{Q_{j,k}}\right)$$

low freq. & horizontal, vertical lines are more important.

Image Compression cont.

Monday, February 13, 2017 10:08 AM

$$\tilde{x}_b[m,n] = \sum_{k=0}^{7} \sum_{j=0}^{7} \tilde{\alpha}_{k,j} \, \psi_{k,j}[m,n] \quad \leftarrow \text{reconstruction}$$

See slides on Color and JPEG for more details