Image Transforms and Image Enhancement in Frequency Domain

Lecture 5, Feb 23th, 2009

Lexing Xie

EE4830 Digital Image Processing
http://www.ee.columbia.edu/~xlx/ee4830/

thanks to G&W website, Mani Thomas, Min Wu and Wade Trappe for slide materials
- Recap for lecture 4
- Observations from HW1
- Changes to HW2
roadmap for today

- 2D-DFT definitions and intuitions
- DFT properties, applications
- pros and cons
- DCT
the return of DFT

- Fourier transform: a continuous signal can be represented as a (countable) weighted sum of sinusoids.

**FIGURE 4.1** The function at the bottom is the sum of the four functions above it. Fourier’s idea in 1807 that periodic functions could be represented as a weighted sum of sines and cosines was met with skepticism.
warm-up brainstorm

- Why do we need image transform?
why transform?

- Better image processing
  - Take into account long-range correlations in space
  - Conceptual insights in spatial-frequency information. what it means to be “smooth, moderate change, fast change, …”
  - Used for denoising, enhancement, restoration, …
- Fast computation: convolution vs. multiplication

- Alternative representation and sensing
  - Obtain transformed data as measurement in radiology images (medical and astrophysics), inverse transform to recover image

- Efficient storage and transmission
  - Energy compaction
  - Pick a few “representatives” (basis)
  - Just store/send the “contribution” from each basis
outline

- why transform
- 2D Fourier transform
  - a picture book for DFT and 2D-DFT
  - properties
  - implementation
  - applications
- discrete cosine transform (DCT)
  - definition & visualization
  - Implementation

next lecture: transform of all flavors, unitary transform, KLT, others ...
1-D continuous FT

- 1D – FT

\[ F(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{-j2\pi \omega x} \]

- 1D – DFT of length N

\[ F(u) = \frac{1}{N} \sum_{n=0}^{N-1} f(n) e^{-j2\pi un} \]

\[ g(\omega x) = e^{-j2\pi \omega x} \]
1-D DFT in as basis expansion

\[ F(u) = \frac{1}{N} \sum_{n=0}^{N-1} f(n) e^{-j2\pi un} \]

Forward transform

\[ y(u) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) a(u, n) \]

Inverse transform

\[ x(n) = \frac{1}{\sqrt{N}} \sum_{u=0}^{N-1} y(u) b(u, n) \]

basis

\[ a(u, n) = e^{-j2\pi \frac{un}{N}} = \cos(2\pi \frac{un}{N}) - j\sin(2\pi \frac{un}{N}) \]
1-D DFT in matrix notations

\[ y(u) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) a(u, n) \]

\[ a(u, n) = e^{-j2\pi \frac{un}{N}} = \cos(2\pi \frac{un}{N}) - jsin(2\pi \frac{un}{N}) \]

\[ u = 0, 1, \ldots, N - 1 \]

\[ y = Ax \]
\[ x = A^{-1}y \]
1-D DFT of different lengths

\[ y = Ax \quad a(u, n) = e^{-j2\pi \frac{un}{N}} \quad u = 0, 1, \ldots, N - 1 \]

\[ x = A^{-1}y = \cos(2\pi \frac{un}{N}) - jsin(2\pi \frac{un}{N}) \]

real(A)  imag(A)

N=8

N=16

N=32

N=64
performing 1D DFT

\[ a(u, n) = e^{-j2\pi \frac{un}{N}} \]

real-valued input

\[ x(n) \]

\[ y(u) = Ax \]

\[ real(y) = real(A) \ast x \]
\[ imag(y) = imag(A) \ast x \]

Note: the coefficients in x and y on this slide are only meant for illustration purposes, which are not numerically accurate.
another illustration of 1D-DFT

$x(n)$

real-valued input

$\text{real}(A'(\cdot))$

$\text{imag}(A'(\cdot))$

$y = Ax$

$y(u) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n)a(u, n)$

$\text{real}(y(u))$

$\text{imag}(y(u))$

Note: the coefficients in x and y are not numerically accurate
from 1D to 2D

1D

signal \( x(n) \)

basis \( a(u, n) \)

\[ e^{-j2\pi \frac{un}{N}} \]

transform coefficients

\[ y(u) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n)a(u, n) \]

matrix form \( y = Ax \)

2D

signal \( x(m, n) \)

basis \( a(u, v, m, n) \)

\[ e^{-j2\pi \left( \frac{um}{N} + \frac{vn}{N} \right)} \]

transform coefficients

\[ y(u, v) = \frac{1}{N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m, n)a(u, v, m, n) \]
Computing 2D-DFT

\[ y(u, v) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m, n) e^{-j\frac{2\pi um}{M}} e^{-j\frac{2\pi vn}{N}} \]

\[ x(m, n) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} y(u, v) e^{j\frac{2\pi um}{M}} e^{j\frac{2\pi vn}{N}} \]

- Discrete, 2-D Fourier & inverse Fourier transforms are implemented in \texttt{fft2} and \texttt{ifft2}, respectively
- \texttt{fftshift}: Move origin (DC component) to image center for display
- Example:
  
  \[
  \begin{align*}
  & >> I = imread('test.png'); \quad \% \text{Load grayscale image} \\
  & >> F = \texttt{fftshift}(\texttt{fft2}(I)); \quad \% \text{Shifted transform} \\
  & >> \texttt{imshow}(\log(\text{abs}(F)),[]); \quad \% \text{Show log magnitude} \\
  & >> \texttt{imshow}(\text{angle}(F),[]); \quad \% \text{Show phase angle}
  \end{align*}
  \]
2-D Fourier basis

\[
\text{real} \quad e^{-j2\pi\left(\frac{um}{N} + \frac{vn}{N}\right)} \\
\text{imag} \quad e^{-j2\pi\left(\frac{um}{N} + \frac{vn}{N}\right)}
\]
2-D FT illustrated

\[ a(u, v, m, n) = e^{-j2\pi\left(\frac{um}{N} + \frac{vn}{N}\right)} \]

real-valued
\[ x(m, n) \]

real
\[ y(u, v) \]

imag
\[ \text{real}(y(u, v)) \]

imag
\[ \text{imag}(y(u, v)) \]
notes about 2D-DFT

- Output of the Fourier transform is a complex number
  - Decompose the complex number as the magnitude and phase components
- In Matlab: \( u = \text{real}(z), \ v = \text{imag}(z), \ r = \text{abs}(z), \ \text{and} \ \theta = \text{angle}(z) \)

<table>
<thead>
<tr>
<th>Function</th>
<th>Formula</th>
<th>Inverse Fourier Transform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Impulse</td>
<td>( \delta(x, y) \leftrightarrow 1 )</td>
<td></td>
</tr>
<tr>
<td>Gaussian</td>
<td>( A\sqrt{2\pi}e^{-2\pi^2\sigma^2(x^2+y^2)} \leftrightarrow Ae^{-(u^2+v^2)/2\sigma^2} )</td>
<td></td>
</tr>
<tr>
<td>Rectangle</td>
<td>( \text{rect}[a, b] \leftrightarrow ab \frac{\sin(\pi ua)}{(\pi ua)} \frac{\sin(\pi vb)}{(\pi vb)} e^{j\pi(ua+vb)} )</td>
<td></td>
</tr>
<tr>
<td>Cosine</td>
<td>( \cos(2\pi u_0 x + 2\pi v_0 y) \leftrightarrow \frac{1}{2} [\delta(u + u_0, v + v_0) + \delta(u - u_0, v - v_0)] )</td>
<td></td>
</tr>
<tr>
<td>Sine</td>
<td>( \sin(2\pi u_0 x + 2\pi v_0 y) \leftrightarrow j\frac{1}{2} [\delta(u + u_0, v + v_0) - \delta(u - u_0, v - v_0)] )</td>
<td></td>
</tr>
</tbody>
</table>

\(^\dagger\) Assumes that functions have been extended by zero padding.
Explaining 2D-DFT

\[ f(x) \]

\[ \frac{AK}{M} \]

\[ 2AK \]

\[ M \text{ points} \]

\[ M \text{ points} \]

\[ M \text{ points} \]
observation 1: compacting energy

**FIGURE 4.11** (a) An image of size $500 \times 500$ pixels and (b) its Fourier spectrum. The superimposed circles have radii values of 5, 15, 30, 80, and 230, which enclose 92.0, 94.6, 96.4, 98.0, and 99.5% of the image power, respectively.

**FIGURE 4.12** (a) Original image, (b)–(f) Results of ideal lowpass filtering with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). The power removed by these filters was 8, 5.4, 3.6, 2, and 0.5% of the total, respectively.
The Phase of DFT

\[ a(u, n) = e^{-j2\pi \frac{un}{N}} = \cos(2\pi \frac{un}{N}) - j\sin(2\pi \frac{un}{N}) \]
intuition of the FFT phase

- Amplitude: relative prominence of sinusoids
- Phase: relative displacement of sinusoids
another example: amplitude vs. phase

A = “Aron”
FA = fft2(A)

P = “Phyllis”
FP = fft2(P)

log(abs(FA))
log(abs(FP))

angle(FA)
angle(FP)

ifft2(abs(FA), angle(FP))
ifft2(abs(FP), angle(FA))

Adpated from http://robotics.eecs.berkeley.edu/~sastry/ee20/vision2/vision2.html
observation 2: amplitude vs. phase

FIGURE 4.27 (a) Woman. (b) Phase angle. (c) Woman reconstructed using only the phase angle. (d) Woman reconstructed using only the spectrum. (e) Reconstruction using the phase angle corresponding to the woman and the spectrum corresponding to the rectangle in Fig. 4.24(a). (f) Reconstruction using the phase of the rectangle and the spectrum of the woman.
fast implementation of 2-D DFT

- 2 Dimensional DFT is separable

\[
F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(m, n) e^{-\frac{2\pi jum}{M}} e^{-\frac{2\pi jvn}{N}}
\]

\[
= \frac{1}{M} \sum_{x=0}^{M-1} e^{-\frac{2\pi jum}{M}} \cdot \frac{1}{N} \sum_{y=0}^{N-1} f(m, n) e^{-\frac{2\pi jvn}{N}}
\]

- 1D FFT: \(O(N \log_2 N)\)
- 2D DFT naïve implementation: \(O(N^4)\)
- 2D DFT as 1D FFT for each row and then for each column: \(O(N^2 \log_2 N)\)
Implement IDFT as DFT

\[ F(u, v) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) e^{-j2\pi\left(\frac{um}{M} + \frac{vn}{N}\right)} \]

\[ f(m, n) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi\left(\frac{um}{M} + \frac{vn}{N}\right)} \]

\[ f^*(m, n) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F^*(u, v) e^{-j2\pi\left(\frac{um}{M} + \frac{vn}{N}\right)} \]

\[ = (MN) \cdot DFT2[F^*(u, v)] \]
# Properties of 2D-DFT

<table>
<thead>
<tr>
<th>Property</th>
<th>Expression(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fourier transform</td>
<td>[ F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)e^{-j2\pi(ux/M + vy/N)} ]</td>
</tr>
<tr>
<td>Inverse Fourier transform</td>
<td>[ f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v)e^{j2\pi(ux/M + vy/N)} ]</td>
</tr>
<tr>
<td>Polar representation</td>
<td>[ F(u, v) =</td>
</tr>
<tr>
<td>Spectrum</td>
<td>[</td>
</tr>
<tr>
<td>Phase angle</td>
<td>[ \phi(u, v) = \tan^{-1}\left( \frac{I(u, v)}{R(u, v)} \right) ]</td>
</tr>
<tr>
<td>Power spectrum</td>
<td>[ P(u, v) =</td>
</tr>
<tr>
<td>Average value</td>
<td>[ \bar{f}(x, y) = F(0, 0) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) ]</td>
</tr>
<tr>
<td>Translation</td>
<td>[ f(x, y)e^{j2\pi(ux/M + vy/N)} \Leftrightarrow F(u - u_0, v - v_0) ]</td>
</tr>
<tr>
<td></td>
<td>[ f(x - x_0, y - y_0) \Leftrightarrow F(u, v)e^{j2\pi(ux/M + vy/N)} ]</td>
</tr>
<tr>
<td></td>
<td>When ( x_0 = u_0 = M/2 ) and ( y_0 = v_0 = N/2 ), then [ f(x, y)(-1)^{x+y} \Leftrightarrow F(u - M/2, v - N/2) ]</td>
</tr>
<tr>
<td></td>
<td>[ f(x - M/2, y - N/2) \Leftrightarrow F(u, v)(-1)^{u+v} ]</td>
</tr>
<tr>
<td>Name</td>
<td>DFT Pairs</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>1) Symmetry properties</td>
<td>See Table 4.1</td>
</tr>
<tr>
<td>2) Linearity</td>
<td>$af_1(x, y) + bf_2(x, y) \Leftrightarrow aF_1(u, v) + bF_2(u, v)$</td>
</tr>
<tr>
<td>3) Translation (general)</td>
<td>$f(x, y)e^{j2\pi(ux/M + vy/N)} \Leftrightarrow F(u - u_0, v - v_0)$</td>
</tr>
<tr>
<td></td>
<td>$f(x + x_0, y + y_0) \Leftrightarrow F(u, v)e^{-j2\pi(ux/M + vy/N)}$</td>
</tr>
<tr>
<td>4) Translation to center of the frequency rectangle, $(M/2, N/2)$</td>
<td>$f(x, y)(-1)^{x+y} \Leftrightarrow F(u - M/2, v - N/2)$</td>
</tr>
<tr>
<td></td>
<td>$f(x - M/2, y - N/2) \Leftrightarrow F(u, v)(-1)^{x+y}$</td>
</tr>
<tr>
<td>5) Rotation</td>
<td>$f(r, \theta + \theta_0) \Leftrightarrow F(\omega, \varphi + \theta_0)$</td>
</tr>
<tr>
<td></td>
<td>$x = r \cos \theta \quad y = r \sin \theta \quad u = \omega \cos \varphi \quad v = \omega \sin \varphi$</td>
</tr>
<tr>
<td>6) Convolution theorem(^\dagger)</td>
<td>$f(x, y) \star h(x, y) \Leftrightarrow F(u, v)H(u, v)$</td>
</tr>
<tr>
<td></td>
<td>$f(x, y)h(x, y) \Leftrightarrow F(u, v) \star H(u, v)$</td>
</tr>
</tbody>
</table>

*(Continued)*
<table>
<thead>
<tr>
<th>Property</th>
<th>Expression(s)</th>
</tr>
</thead>
</table>
| Computation of the inverse Fourier transform using a forward transform algorithm | \[
\frac{1}{MN} f^*(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F^*(u, v) e^{-j2\pi(ux/M + vy/N)}
\]
This equation indicates that inputting the function $F^*(u, v)$ into an algorithm designed to compute the forward transform (right side of the preceding equation) yields $f^*(x, y)/MN$. Taking the complex conjugate and multiplying this result by $MN$ gives the desired inverse. |
| Convolution†                                                             | \[
(f \ast h)(x, y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n)h(x - m, y - n)
\] |
| Correlation†                                                             | \[
(f \circ h)(x, y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f^*(m, n)h(x + m, y + n)
\] |
| Convoluted theorem†                                                      | $f(x, y) \ast h(x, y) \Leftrightarrow F(u, v)H(u, v)$; $f(x, y)h(x, y) \Leftrightarrow F(u, v) \ast H(u, v)$ |
| Correlation theorem†                                                     | $f(x, y) \circ h(x, y) \Leftrightarrow F^*(u, v)H(u, v)$; $f^*(x, y)h(x, y) \Leftrightarrow F(u, v) \circ H(u, v)$ |
outline

- why transform
- 2D Fourier transform
  - a picture book for DFT and 2D-DFT
  - properties
  - implementation
  - applications
- discrete cosine transform (DCT)
  - definition & visualization
  - implementation
DFT application #1: fast Convolution

Frequency domain filtering operation

[Diagram showing the process of spatial filtering: pre-processing, Fourier transform, filter function $H(u, v)$, inverse Fourier transform, post-processing.]

$O(N^2)$

Spatial filtering
$f(x, y) * h(x, y)$

Enhanced image
DFT application #1: fast convolution

Spatial filtering

\[ f(x,y) \ast h(x,y) \]

\[ O(N^4) \]

Frequency domain filtering operation

- Pre-processing: \[ O(N^2 \cdot \log_2 N) \]
- Fourier transform: \[ F(u,v) \]
- Filter function: \[ H(u,v) \]
- Inverse Fourier transform: \[ H(u,v)F(u,v) \]
- Post-processing: \[ g(x,y) \]

\[ O(N^2) \]

\[ O(N^2 \cdot \log_2 N) \]
DFT application #2: feature correlation

Find letter “a” in the following image

```
bw = imread('text.png'); a = imread('letter_a.png');
% Convolution is equivalent to correlation if you rotate the
% convolution kernel by 180deg
C = real(ifft2(fft2(bw) .*fft2(rot90(a,2),256,256)));

% Use a threshold that's a little less than max.
% Display showing pixels over threshold.
thresh = .9*max(C(:)); figure, imshow(C > thresh)
```

from Matlab image processing demos.
DFT application #3: image filters

- A zoology of image filters
  - Smoothing / Sharpening / Others
  - Support in time/space vs. support in frequency c.f. “FIR / IIR”
  - Definition: spatial domain/frequency domain
  - Separable / Non-separable
circular convolution and zero padding
Zero-padded filter and response

- Zero-padding avoids wrapping error in filters

- It also increases frequency-domain resolution, i.e., interpolation in the frequency domain.
  - A given spatial domain signal has a fixed spatial resolution, e.g., $G_0 = 75$ dpi (dots-per-inch).
  - The highest spatial frequency that this signal can represent is $F_0 = 37.5$ cycles per inch, due to Nyquist theorem.
  - $F_0$ maps to digital frequency $\omega_0 = \pi$, and an $N_0$-75 point DFT gives frequency resolution of $\pi/F_0 = 0.084$ rad
  - Zero-padded signal increases the frequency resolution of the transformed signal, e.g., padding zero and taking $N_1 = 100$ DFT gives a resolution of $\pi/50 = 0.0628$

- Note that zero-padding does NOT introduce new information, NEITHER does it increase the maximum frequency.
zero padded filter and response

**FIGURE 4.39** Padded lowpass filter is the spatial domain (only the real part is shown).

**FIGURE 4.40** Result of filtering with padding. The image is usually cropped to its original size since there is little valuable information past the image boundaries.
smoothing filters: ideal low-pass

FIGURE 4.10 (a) Perspective plot of an ideal lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.
butterworth filters

\[ H(u, v) = \frac{1}{1 + [D(u, v)/D_0]^{2n}} \]

**FIGURE 4.14** (a) Perspective plot of a Butterworth lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.

**FIGURE 4.16** (a)-(d) Spatial representation of BLPFs of order 1, 2, 5, and 20, and corresponding gray-level profiles through the center of the filters (all filters have a cutoff frequency of 5). Note that ringing increases as a function of filter order.
Gaussian filters

\[ H(u, v) = e^{-D^2(u,v)/2\sigma^2} \]

**FIGURE 4.17** (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of \( D_0 \).
low-pass filter examples

FIGURE 4.18 (a) Original image. (b)–(f) Results of filtering with Gaussian lowpass filters with cutoff frequencies set at radii values of 5, 15, 30, 60, and 200, as shown in Fig. 4.11(b). Compare with Figs. 4.12 and 4.15.

FIGURE 4.15 (a) Original image. (b)–(f) Results of filtering with BLIPs of order 2, with cutoff frequencies at radii of 5, 15, 30, 60, and 200, as shown in Fig. 4.11(b). Compare with Fig. 4.12.

FIGURE 4.12 (a) Original image. (b)–(f) Results of ideal lowpass filtering with cutoff frequencies set at radii values of 5, 15, 30, 60, and 200, as shown in Fig. 4.11(b). The power removed by these filters was 8, 5, 4, 3, and 1% of the total, respectively.
smoothing filter application 1

text enhancement

FIGURE 4.19
(a) Sample text of poor resolution (note broken characters in magnified view).
(b) Result of filtering with a GLPF (broken character segments were joined).

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company’s software may recognize a date using “00” as 1900 rather than the year 2000.

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company’s software may recognize a date using “00” as 1900 rather than the year 2000.
smoothing filter application 2

beautify a photo

FIGURE 4.20 (a) Original image (1024 × 732 pixels). (b) Result of filtering with a GLPF with $D_0 = 100$. (c) Result of filtering with a GLPF with $D_0 = 80$. Note reduction in skin fine lines in the magnified sections of (b) and (c).
high-pass filters

\[ H_{HPF}(u, v) = 1 - H_{LPF}(u, v) \]

**FIGURE 4.22** Top row: Perspective plot, image representation, and cross section of a typical ideal highpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian highpass filters.
sobel operator in frequency domain

Question:
Sobel vs. other high-pass filters?
Spatial vs frequency domain implementation?
high-pass filter examples

**FIGURE 4.24** Results of ideal highpass filtering the image in Fig. 4.11(a) with $D_0 = 15$, 30, and 80, respectively. Problems with ringing are quite evident in (a) and (b).

**FIGURE 4.26** Results of highpass filtering the image of Fig. 4.11(a) using a GHPF of order 2 with $D_0 = 15$, 30, and 80, respectively. Compare with Figs. 4.24 and 4.25.
band-pass, band-reject filters

FIGURE 4.64
(a) Sampled newspaper image showing a moiré pattern.
(b) Spectrum.
(c) Butterworth notch reject filter multiplied by the Fourier transform.
(d) Filtered image.
outline

- why transform
- 2D Fourier transform
  - a picture book for DFT and 2D-DFT
  - properties
  - implementation
  - applications in enhancement, correlation
- discrete cosine transform (DCT)
  - definition & visualization
  - implementation
Is DFT a Good (enough) Transform?

- Theory
- Implementation
- Application
The Desirables for Image Transforms

- **Theory**
  - Inverse transform available
  - Energy conservation (Parsevell)
  - Good for compacting energy
  - Orthonormal, complete basis
  - (sort of) shift- and rotation invariant

- **Implementation**
  - Real-valued
  - Separable
  - Fast to compute w. butterfly-like structure
  - Same implementation for forward and inverse transform

- **Application**
  - Useful for image enhancement
  - Capture perceptually meaningful structures in images

<table>
<thead>
<tr>
<th>DFT</th>
<th>???</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>?</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>
DCT defined w.r.t DFT

\[ y = Ax \]

1D-DCT

\[
a(0, n) = \sqrt{\frac{1}{N}} \quad u = 0
\]

\[
a(u, n) = \sqrt{\frac{2}{N}} \cos\left(\frac{\pi(2n + 1)u}{2N}\right) \quad u = 1, 2, \ldots, N - 1
\]

1D-DFT

\[
a(u, n) = e^{-j2\pi\frac{un}{N}} = \cos(2\pi\frac{un}{N}) + j\sin(2\pi\frac{un}{N})
\]
1-D Discrete Cosine Transform (DCT)

\[
\begin{aligned}
Z(k) &= \sum_{n=0}^{N-1} z(n) \cdot \alpha(k) \cos \left[ \frac{\pi(2n+1)k}{2N} \right] \\
z(n) &= \sum_{k=0}^{N-1} Z(k) \cdot \alpha(k) \cos \left[ \frac{\pi(2n+1)k}{2N} \right] \\
\alpha(0) &= \frac{1}{\sqrt{N}}, \alpha(k) = \sqrt{\frac{2}{N}}
\end{aligned}
\]

- Transform matrix \( A \)
  - \( a(k,n) = \alpha(0) \) for \( k=0 \)
  - \( a(k,n) = \alpha(k) \cos[\pi(2n+1)/2N] \) for \( k>0 \)

- \( A \) is real and orthogonal
  - rows of \( A \) form orthonormal basis
  - \( A \) is not symmetric!
  - DCT is not the real part of unitary DFT!
1-D DCT

Original signal

Transform coeff.

Basis vectors

Reconstructions
DFT and DCT in Matrix Notations

Matrix notation for 1D transform

\[ y = Ax, \quad x = A^{-1}y \]

1D-DCT
\[
a(0, n) = \sqrt{\frac{1}{N}} \quad u = 0
\]
\[
a(u, n) = \sqrt{\frac{2}{N}} \cos\left(\frac{\pi(2n + 1)u}{2N}\right) \quad u = 1, 2, \ldots, N - 1
\]

N=32

1D-DFT
\[
a(u, n) = e^{-j2\pi \frac{un}{N}} = \cos(2\pi \frac{un}{N}) - jsin(2\pi \frac{un}{N})
\]

A
real(A)
imag(A)
From 1D-DCT to 2D-DCT

- Rows of $A$ form a set of orthonormal basis
- $A$ is not symmetric!
- DCT is not the real part of unitary DFT!
basis images: DFT (real) vs DCT
Periodicity Implied by DFT and DCT

**FIGURE 8.32** The periodicity implicit in the 1-D (a) DFT and (b) DCT.
DFT and DCT on Lena

DFT2

Shift low-freq to the center

Assume periodic and zero-padded ...

DCT2

Assume reflection ...

Using FFT to implement fast DCT

- Reorder odd and even elements
  \[
  \begin{align*}
  \tilde{z}(n) &= z(2n) \\
  \tilde{z}(N-n-1) &= z(2n+1)
  \end{align*}
  \]
  for \(0 \leq n \leq \frac{N}{2} - 1\)

- Split the DCT sum into odd and even terms
  \[
  Z(k) = \alpha(k) \left\{ \sum_{n=0}^{N/2-1} z(2n) \cdot \cos \left( \frac{\pi (4n + 1) k}{2N} \right) + \sum_{n=0}^{N/2-1} z(2n + 1) \cdot \cos \left( \frac{\pi (4n + 3) k}{2N} \right) \right\}
  \]
  \[
  = \alpha(k) \left\{ \sum_{n=0}^{N/2-1} \tilde{z}(n) \cdot \cos \left( \frac{\pi (4n + 1) k}{2N} \right) + \sum_{n=0}^{N/2-1} \tilde{z}(N-n-1) \cdot \cos \left( \frac{\pi (4n + 3) k}{2N} \right) \right\}
  \]
  \[
  = \alpha(k) \left\{ \sum_{n=0}^{N/2-1} \tilde{z}(n) \cdot \cos \left( \frac{\pi (4n + 1) k}{2N} \right) + \sum_{n'=N/2}^{N-1} \tilde{z}(n') \cdot \cos \left( \frac{\pi (4N - 4n'-1) k}{2N} \right) \right\}
  \]
  \[
  = \alpha(k) \sum_{n=0}^{N-1} \tilde{z}(n) \cdot \cos \left( \frac{\pi (4n + 1) k}{2N} \right) = \text{Re} \left[ \alpha(k) e^{-j\pi k / 2N} \sum_{n=0}^{N-1} \tilde{z}(n) \cdot e^{-j2\pi nk / N} \right]
  \]
  \[
  = \text{Re} \left[ \alpha(k) e^{-j\pi k / 2N} \text{DFT} \ \{\tilde{z}(n)\}_N \right]
  \]
# The Desirables for Image Transforms

<table>
<thead>
<tr>
<th>Theory</th>
<th>DFT</th>
<th>DCT</th>
<th>???</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inverse transform available</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Energy conservation (Parsevell)</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Good for compacting energy</td>
<td>?</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>Orthonormal, complete basis</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>(sort of) shift- and rotation invariant</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Implementation</th>
<th>DFT</th>
<th>DCT</th>
<th>???</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real-valued</td>
<td>x</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Separable</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Fast to compute w. butterfly-like structure</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Same implementation for forward and inverse transform</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Application</th>
<th>DFT</th>
<th>DCT</th>
<th>???</th>
</tr>
</thead>
<tbody>
<tr>
<td>Useful for image enhancement</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Capture perceptually meaningful structures in images</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Summary of Lecture 5

- Why we need image transform
- DFT revisited
  - Definitions, properties, observations, implementations, applications
- What do we need for a transform
- DCT

- Coming in Lecture 6:
  - Unitary transforms, KL transform, DCT
  - examples and optimality for DCT and KLT, other transform flavors, Wavelets, Applications

- Readings: G&W chapter 4, chapter 5 of Jain has been posted on Courseworks

- "Transforms" that do not belong to lectures 5-6: Rodon transform, Hough transform, ...
"Given a signal with a 2-Hz bandwagon, the Nyquist frequency must be 1-Hz to avoid aliasing of the Fourier transformer."

I take it that's wrong?

Wrong? It's downright pathetic! 1-Hz!? Come on!