Announcement (cont’d)

- Generate two cropped & downsampled face images
  - A little extra work for Part-II 7
  - For use in next labs
  - Face image → matrix representation
  - Crop facial part by selecting the corresponding part in the matrix
  - Matlab function for resizing “imresize”
  - Obtain a 128x128 and a 32x32 face image
  - Write into a JPEG image with default quality factor
  - Put the original and the two new one on webpage

Review of Last Class

- Vector/matrix representation of 1-D & 2-D sampled signal
  - Representing an image as a matrix or sometimes as a long vector
- Basis functions/vectors and orthonormal basis
  - Used for representing the space via their linear combinations
  - Many possible sets of basis and orthonormal basis
- Unitary transform on input $\mathbf{x}$ → $A^T = A^*$
  - $y(i) = \sum a_i y(i)$ – represented by basis vectors $\{a_i\}$
  - Rows (and columns) of a unitary matrix form an orthonormal basis
- General 2-D transform and separable unitary 2-D transform
  - 2-D transform involves $O(N^4)$ computation
  - Separable: $Y = AXA^*$ → $(AX)A^* = O(N^3)$ computation
  - Apply 1-D transform to all columns, then apply 1-D transform to rows
**Warm-up Exercises**

- **Unitary or not?**
  - Find basis for unitary one
    
- **Find basis images and represent image \( X \) with basis images**
  - \( X = A^H Y A^* \) (separable)
  - \( x(m,n) = \sum_k \sum_l a^*(k,m)a^*(l,n) y(k,l) \)

- Represent \( X \) with \( N \times N \) basis images weighted by coeff. \( Y \)

- Obtain basis image \( \{ a^*(k_0,m)a^*(l_0,n) \}_{m,n} \) by setting \( Y = \{ \delta(k-k_0,l-l_0) \} \) & getting \( X \)

- In matrix form \( A^*_k l = a^*_k a^*_l \)\(^T\)

- \( a^*_k \) is \( k \)th column vector of \( A^* \)

- \( a_k \) is \( k \)th row vector of \( A \)

- Transf. coeff. \( y(k,l) \) is the inner product of \( A^*_k l \) with the image

*Jain’s e.g.5.1, pp137

\[
A = \begin{bmatrix}
1 & 1 \\
\sqrt{2} & -j \\
1 & -1
\end{bmatrix},
X = \begin{bmatrix}
1 & 2 \\
3 & 4
\end{bmatrix}
\]

**Clarifications**

- **“Dimension”**
  - Dimension of a signal \( \sim \) # of index variables
    - audio and speech is 1-D signal, image is 2-D, video is 3-D
  - Dimension of a vector space \( \sim \) # of vectors in its basis

- **Eigenvalues of unitary transform**
  - All eigenvalues have unit magnitude (could be complex valued)
    - By definition of eigenvalues \( A X = \Lambda X \)
    - By energy perservation of unitary \( ||A X|| = ||X|| \)
  - Eigenvalues here are different from the eigenvalues in K-L transform
    - K-L concerns the eigen of covariance matrix of random vector
  - Eigenvectors \( \sim \) we generally consider the orthonormalized ones

**Overview of Today’s Lecture**

- **Examples of unitary transforms**
  - DFT
  - DCT
  - K-L transform
  - Haar

**1-D DFT with Representation in Unitary Transform**

\[
\{ z(n) \} \Leftrightarrow \{ Z(k) \}
\]

- \( n, k = 0, 1, \ldots, N-1 \)
- \( W_N = \exp\{ -j2\pi/N \} \)
- Complex conjugate of primitive \( N \)th root of unity

- **Basis vectors**
  - \( Z = \sum_k Z(k) \omega \) \( \Rightarrow \) what are the \( \{ \omega_k \} \)?
  - \( \omega_k = [1 \ W_N^k \ W_N^{2k} \ldots W_N^{N-1k}] / \sqrt{N} \)
  - \( z = \sum_k Z(k) \omega_k \)
  - Use \( \omega_k \) as row vectors to construct a matrix \( F \)
  - \( Z = F \tilde{z} \Leftrightarrow \tilde{z} = F^* Z \)
    - \( F \) is symmetric and unitary
2-D DFT

- 2-D DFT is Separable
  - \( Y = F X F^* \)  \( X = F^* Y F \)
  - Basis images: \( B_{kl} = (f_N^n - k)(f_N^m - l) \)

Properties of 2-D DFT

- Conjugate symmetry for real image
  - Recall similar symmetry for 1-D DFT
- \( N^2 \) independent element from input => same independence in output
- 2-D circular convolution vs. multiplication
  - See Jain's book pp147 for more details.

In general, DFT is complex valued

1-D Discrete Cosine Transform (DCT)

- Transform matrix \( C \)
  - \( c(k,n) = \alpha(0) \) for \( k=0 \)
  - \( c(k,n) = \alpha(k) \cos(\pi(2n+1)/2N) \) for \( k>0 \)
- \( C \) is real and orthogonal
  - Rows of \( C \) form orthonormal basis
  - \( C \) is not symmetric!
  - DCT is not the real part of unitary DFT!
    - Related to DFT of a symmetrically extended signal

Example of 1-D DCT

From Ken Lam's DCT talk 2001 (HK Polytech)

Example of 1-D DCT (cont'd)

From Ken Lam's DCT talk 2001 (HK Polytech)
**Fast Transform via FFT**

- Define new sequence
  - reorder odd and even elements
    \[
    \tilde{x}(n) = x(2n), \quad \tilde{x}(N-n-1) = x(2n+1) \quad \text{for } 0 \leq n \leq \frac{N-1}{2}
    \]
- Split DCT sum into odd and even terms

**Other real-value fast algorithms**

\[
\begin{align*}
2 \leq t \leq 0 & \Rightarrow \left( \begin{array}{c}
\lambda_1 = \lambda_2 = \ldots = \lambda_{t-1} = 0 \\
\lambda_t, \lambda_{t+1}, \ldots, \lambda_N
\end{array} \right) \Rightarrow  \\
& = 0
\end{align*}
\]

**2-D DCT**

- Separable orthogonal transform
- \[ Y = C X C^T \quad \Rightarrow \quad X = C^T Y C \]
- DCT basis images

**K-L Transform (Principal Component Analysis)**

- Recall the unanswered question
  - what unitary transform gives the best compaction and decorrelation?
- Consider an \( N \times 1 \) zero-mean random vector \( x \)
  - Covariance (autocorrelation) matrix \( R = E[x x^H] \)
    - give ideas of correlation between elements
- Eigen decomposition of \( R \)
  - eigen vectors \( R \vec{u}_i = \lambda_i \vec{u}_i \)
- K-L transform \( y = U^H x \) with \( U = [ \vec{u}_1, \ldots, \vec{u}_N ] \)
  - Basis vectors of K-L transf. is the orthonormalized eigenvectors of \( R \)
  - Note \( U^H R U = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N) \)
  - For convenience, reorder \( \{\vec{u}_i\} \) so that \( \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N \)

**Properties of K-L Transform**

- **Decorrelation**
  - \( E[ y y^H ] = E[ U^H x x^H ] U^H U = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N) \)
  - Other matrices (unitary or nonunitary) may also decorrelate the transformed sequence (Jain’s e.g.5.7 pp166).
- **Minimum MSE**
  - If only allow to keep \( K \) coefficients for any \( 1 \leq K \leq N \), what’s the best way?
  - Answer in MMSE sense \( \Rightarrow \text{Keep the coefficients w.r.t. the eigenvectors of the first largest eigenvalues} \)
  - **Proof**: Theorem5.1 in Jain’s (pp166)
**K-L Transform for Images**
- Work with 2-D autocorrelation function
  \[ R(m,n; m',n') = E\{ x(m, n) x(m', n') \} \] for all \( 0 \leq m, m', n, n' \leq N-1 \)
  - K-L Basis images is the orthonormalized eigenfunctions of \( R \)
- Rewrite images into vector form (\( N^2x1 \))
  - Need solve the eigen problem for \( N^2xN^2 \) matrix! \( \sim O(N^6) \)
- Reduced computation for separable \( R \)
  \[ R(m,n; m',n') = r_1(m,n) r_2(m',n') \]
  - Only need solve the eigen problem for two \( NxN \) matrices
  \( \sim O(N^3) \)

**Pros and Cons of K-L Transform**
- Optimality
  - Decorrelation and MMSE for the same # of partial coeff.
- Data dependent
  - Have to estimate the 2nd-order statistics to determine the transform
  - Can we get data-independent transf. with similar performance?
    - DCT
- Applications
  - (non-universal) compression
  - pattern recognition: e.g., eigen faces
  - analyze the principal (“dominating”) components

**Energy Compaction of DCT vs. K-L Transform**
- Excellent energy compaction of DCT
  - for highly correlated data
- DCT is close to K-L transf. of 1st-order stationary Markov
  - DCT basis vectors are eigenvectors of a symmetric tridiagonal matrix \( Q_\epsilon \)
  - Covariance matrix \( R \) of 1st-order stationary Markov sequence has an inverse in the form of symmetric tridiagonal matrix
  - For highly correlated sequence, the scaled version of \( R^{-1} \) approx. \( Q_\epsilon \)
  - See Jain’s pp183 for details.
- DCT is a good replacement for K-L
  - Close to optimal for highly correlated data
  - Not depend on specific data like K-L does
  - Fast algorithm available

**Construction of Haar functions**
- Unique decomposition of integer \( k \propto (p, q) \)
  - \( k = 0, \ldots, N-1 \) with \( N = 2^n, 0 \leq p \leq n-1 \)
  - \( q = 0, 1 \) (for \( p=0 \)); \( 1 \leq q \leq 2^p \) (for \( p>0 \))
  - e.g., \( k=0 \propto (0,0), k=1 \propto (0,1); k=2 \propto (1,1), k=3 \propto (1,2) \)
- \( h_k(x) = h_{p,q}(x) \) for \( x \in [0,1] \)
  - for \( x \in [0,1] \)
    \[ h_k(x) = h_{p,q}(x) = \frac{1}{\sqrt{N}} \left\{ \begin{array}{ll}
    1 & \text{for} \ \frac{g_1}{2^p} \leq x < \frac{2g_1}{2^p} \\
    \frac{1}{\sqrt{N}} 2^{p/2} & \text{for} \ \frac{g_1}{2^p} \leq x < \frac{g_1}{2^p} \\
    0 & \text{for other} \ \ x \in [0,1]
  \end{array} \right. \]
**Haar Transform**
- Haar transform $H$
  - Sample $h_k(x)$ at $\{m/N\}$
  - $m = 0, ..., N-1$
  - Real and orthogonal
  - Transition at each scale $p$ is localized according to $q$
- Basis images of 2-D (separable) Haar transform

**Summary**
- Common unitary transforms
  - 1-D transform and basis vectors
  - 2-D separable and basis images
- DFT
- DCT
  - Real valued
  - Good energy compaction for highly correlated data
- K-L
  - Best energy compaction but data dependent
- Haar
  - Localize transitions

**Preview of Next Time**
- Use as few bits as possible to encode an image
  - Image compression
- Basic tools
  - Lossless tools
  - Lossy tools
- Which domain to work with?
  - Directly with pixels?
  - Will some smart transforms help?

**Assignment**
- Readings
  - Jain’s book 5.4-5.6, 5.9, 5.11
- Reminder
  - Assignment-1 Due Wed. 9/19 11:59pm
  - New addition to Part-II
  - Hand-in writeup
  - Put images and computer codes online
  - Thurs. class will be in Jasmine.