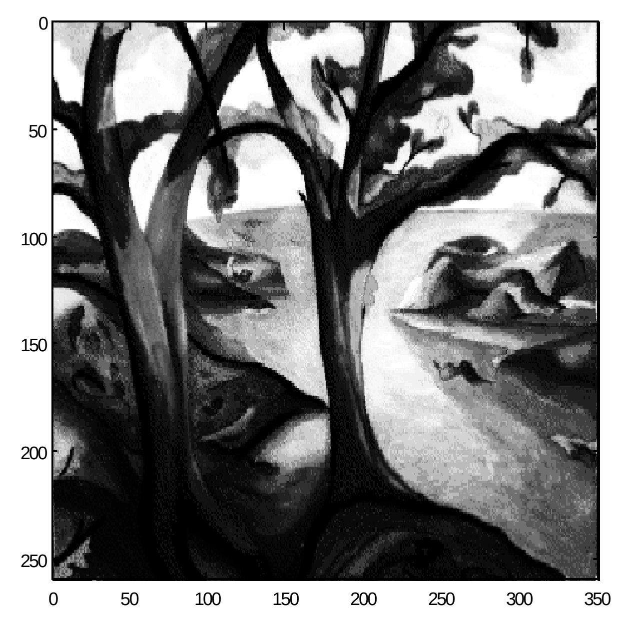

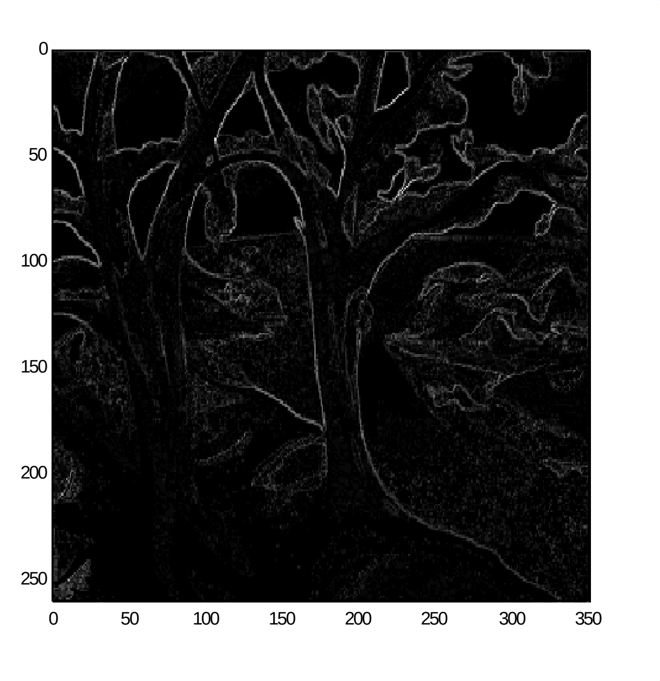

Transform Methods in Image Processing

Mount Holyoke Mathematics Seminar

S. Allen Broughton - Spring Term, 2001

http://www.rose-hulman.edu/~brought/Epubs/mhc/mhctransimage.html

Background - 1

- based on image processing and compression courses taught for several years
- taught to juniors & seniors in math, computer science, physics/applied optics, and electrical/computer engineering
- course for Imaging Systems Certificate
- collaboration of faculty from above departments

Background - 2

- matrix algebra and Fourier series base, DSP helpful
- can get by with matrix algebra only
- Matlab heavily used, Maple to a lesser extent
- course notes under continuous revision
- http://www.rose-hulman.edu/~brought/courses/ma490mip/

Outline of Talks

- ***Lecture 1***- Introduction, various transforms
  - Discrete Fourier Transform DFT and FFT
  - Discrete Cosine transform DCT (pictures only)
  - Discrete Wavelet transform DWT (pictures only)
  - motivating application: motion fields by digital image correlation
- ***Lecture 2***- Convolution/Filtering, Filter Banks to Wavelets
  - filtering and convolution
  - DIC application
  - filter banks
  - motivating application: JPEG and fingerprint image compression

Lecture 1 - 1

- Some image processing problems
- MATLAB demos
- signal and image models
- time domain representation of signals and images
- colour vs black and white
- vector space models
- transforms and processing

Lecture 1 - 2

- frequency domain representation
- Discrete Fourier Transform (DFT) of signals and images
- matrix representation
- analysis and synthesis waveforms
- computational issues

Image Processing Problems

- ***restoration*****:** deblurring photos, e.g., Hubble telescope images
- ***edge detection:*** medical imaging
- ***denoising:*** part of restoration
- ***compression:*** FBI finger print problem, JPEG image storage, transmission, and retrieval
- ***digital image correlation:*** detecting motion with before and after photos

Matlab Examples

- restoration - ansmid3.m
- edge detection - edgedet.m
- denoising - dftdemo.m

Image and Signal Models - 1

- A (general) signal is a function on a time or spatial domain (or combination) $D=D_t$

$$X : D_t \text{ ® } \mathbf{R} \text{ or } X : D_t \text{ ® } \mathbf{C},$$

- in general

$$X \text{ Î } L^2(D_t)$$

- energy of a signal is (proportional to) $X \cdot X$ Hermitian inner product

Image and Signal Models - 2

- continuous models (domain is continuous)
  - audio: $D_t = T$ (real line, time)
  - audio: $D_t = T$ (interval on real line, time, periodic signal)
  - image: $D_t = R$ (rectangle in plane)
  - movie: $D_t = R \times T$ (spatial × time)
  - scientific data: $D_t = V$ (3D volume)
  - scientific data: D_t can be quite an arbitrary set

Image and Signal Models - 3

- discrete models
  - audio: $D_t = \boldsymbol{Z}$ (integers, bi-infinite, uniform discrete sampling)
  - audio: $D_t = \boldsymbol{Z}_N$ (integers mod N, finite, discrete sampling, periodic signal, wrap around)
  - image: $D_t = \boldsymbol{Z}_m \times \boldsymbol{Z}_n$ (rectangle in plane, computer screen with wrap around scrolling)
  - movie: $D_t = (\boldsymbol{Z}_m \times \boldsymbol{Z}_n) \times \boldsymbol{Z}_N$

Image and Signal Models - 4

- continuous models
  - used for analysis and modeling physical processes
  - more difficult for students on the first pass, need calculus/real analysis
  - discretize for computation
- discrete models
  - concentrate on finite duration, discretely sampled signals and images
  - theory simpler, need only linear algebra
  - well adapted for computation

Image and Signal Models - 5

- signals are vectors in $\boldsymbol{R}^N$ or $\boldsymbol{C}^N = L^2(\mathbf{Z}_N)$, as a convolution algebra
- images are matrices in $M_{\boldsymbol{R}}(m,n)$ or $M_{\boldsymbol{C}}(m,n) = L^2(\mathbf{Z}_m \times \mathbf{Z}_n)$, as a convolution algebra
- signals and images are also quantized (non-linear process)
  - 8 bit sound has 256 values
  - 24 bit colour has 3 colours at 256 levels each

Colour

- 8-bit pseudo-colour
  - 256 color levels
  - colour lookup table
- 24 bit true colour
  - three colour matrices *R*, *G*, *B*
  - 256 levels for each colour
- assume images are monochrome for simplicity
- show MATLAB example - eightbit.m

Vector & Matrix Models

- good model for (additive) noise
  - $Y=X+E$ (signal + noise)
- superposition of basic signals and images is a vector space process
- image processing as a linear operator
  - X ® AX (signals)
  - X ® AXB (images)
  - A and B are matrices of the appropriate size

General Transform Process

$$X \xrightarrow{\quad T \quad} \tilde{X} \xrightarrow{\quad process \quad} \tilde{Y} \xrightarrow{\quad T^{-1} \quad} Y$$

- T is some linear transform
- the middle process is usually non-linear
- examples of the middle process are quantization (for compression), deblurring. truncation (for denoising)
- Matlab demo dftdemo.m
- Matlab demo ansmid3.m again

Frequency Domain Representation/Transforms - 1

- Now comes the design part of the transform process
  1. select some set of parameters of interest, they will form a parameter space D_f
  2. for audio signals this is called the frequency domain and the parameter selected is frequency
  3. for each point $\omega \in D_f$ in the frequency domain pick a ***pure wave form*** E_w that typifies this parameter value

Frequency Domain Representation/Transforms - 2

4. Define $\hat{X} : D_f \to \mathrm{C}$

by $\hat{X}(w) = X \cdot E_w$

5. $\hat{X}(w)$ is interpreted as the energy content at the frequency ω

6. $\hat{X}$ is the frequency domain representation of the signal X or the ???? transform of X

7. $T : X \to \hat{X}$ is the ???? transform

8. $T^{-1} : \hat{X} \to X$ is the inverse ???? transform

Frequency Domain Representation/Transforms - 3

- Often the transform parameters are created in the following way

1. There is a lie group action on the space of signals derived, in part from a geometric action on the time or spatial domain, indeed the domain may be a group
2. The frequency domain is the space of irreducible representations (Fourier transform) or may be the group (continuous wavelet transform, windowed Fourier transform)
3. pure wave forms may be irreducible characters (Fourier transform) or nicely constructed signals on the space using the group action

Frequency Domain Representation/Transforms - 4

- Example 1: standard Fourier transform
  - X is a signal on the reals $D_t = R$
  - frequency domain $D_f = R$
  - $E_w = exp(i\,wt)$

$$\hat{X}(w) = \int_{-\infty}^{\infty} X(t) e^{-iwt} dt$$

Frequency Domain Representation/Transforms - 5

- Example 2: Discrete Fourier Transform of a 1D signal
  - $X = [X(0), \ldots, X(N\text{-}1)]^t$ a finite discretely sampled signal on $D_t = \boldsymbol{Z}_N$
  - frequency domain $D_f = \boldsymbol{Z}_N$
  - $E_{N,k}(r)$ = exp$(2pi\ kr/N)$, r Î $\boldsymbol{Z}_N$

$$\hat{X}(k) = X \cdot E_{N,k} = \sum_{r=0}^{N-1} X(r)\exp(-2pi(k \cdot \frac{r}{N}))$$

- Matlab demo dft1demo3

Frequency Domain Representation/Transforms - 6

- Example 3: Discrete Fourier Transform of an image
  - $X = [X(i,j)]$ is an $m \times n$ image in matrix form $D_t = \mathbf{Z}_m \times \mathbf{Z}_n$
  - frequency domain $D_f = \mathbf{Z}_m \times \mathbf{Z}_n$ (vertical, horizontal) frequencies
  - $E_{m,n,k,l} = E_{m,k} (E_{n,l})^t$ as matrices
  - Matlab demo wavefft2cband.m (polar decomposition)

Frequency Domain Representation/Transforms - 7

- Example 4: Windowed Fourier transform of a signal
  - X is a signal on R
  - g is a windowing function of compact support or a Gaussian, of norm 1
  - $G = R^2$ acts via $(p,q)\cdot X = e^{iqt}X(t-p)$
  - pure wave forms are $e^{iqt}g(t-p)$

$$\hat{X}(p,q) = \int_{-\infty}^{\infty} X(t)e^{-iqt}g(t-p)dt$$

Frequency Domain Representation/Transforms - 8

- Example 5: Continuous Wavelet transform of a signal
    - X is a signal on R
    - y is an appropriate mother wavelet function, of norm1, compact support or a derivative of a Gaussian
    - $G = ax+b$ group acts via $(a,b)\cdot X = X((t-b)/a)$
    - pure wave forms are $a^{-1/2}y((t-b)/a)$

$$\hat{X}(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} X(t) y (\frac{t-b}{a}) dt$$

Frequency Domain Representation/Transforms - 9

- Maple worksheet wft.mws
- Maple worksheet cwt.mws

Matrix Representation of DFT -1

- $(E_k)^*$ is the Hermitian transpose of E_k
- as a matrix product $\hat{X}(k) = E_k^* X$
- Set $F_N = [E_0, \ldots, E_{N-1}]^*$

$$\hat{X} = F_N X$$

- $F_N(r,s) = z^{rs}$, $z=exp(-i/N)$

- $(F_N)^* F_N = N I_N$ thus we have the inversion formula

$$X = \frac{1}{N} F_N^* \hat{X}$$

Matrix Representation of DFT -2

- The 2D DFT of an image is given by doing 1D transforms to all columns and then all rows

$$X \circledR F_m X(F_n)^t = (F_m X)(F_n)^t$$

Analysis and Synthesis Waveforms - 1

- Suppose $Y= MX$ is any matrix transform on 1D signals
- Inverse transforms is $X=M^{-1}Y$
- $E_0 , \ldots, E_{N-1}$ columns of M^t (transposed rows of M), are the ***analysis or decomposition waveforms***
- “Fourier” coefficients of X are given by the matrix product $(E_r)^t X$

Analysis and Synthesis Waveforms - 2

- $F_0, \ldots, F_{N-1}$ *columns of* M^{-1} , are the ***synthesis or reconstruction waveforms***
- the original signal

$$X = M^{-1} Y = [F_0, \ldots, F_{N-1}]Y$$

- is a linear combination of the synthesis waveforms and the coefficients are the “Fourier” coefficients

Analysis and Synthesis Waveforms - 3

- Matlab demos of analysis and synthesis
  - waveformsDFT.m
  - analsynDCT.m
  - analsynDWT.m

Computational Issues

- DFT, DCT and DWT all have fast computational algorithms
  - fast algorithm for DFT is known as FFT (Cooley-Tukey algorithm)
  - sample at 2^b points to get speed
  - performance in $O(Nlog(N))$ not $O(N^3)$
  - Matlab demo fftperform.m

End of Lecture 1

Questions, Questions, Questions!!!