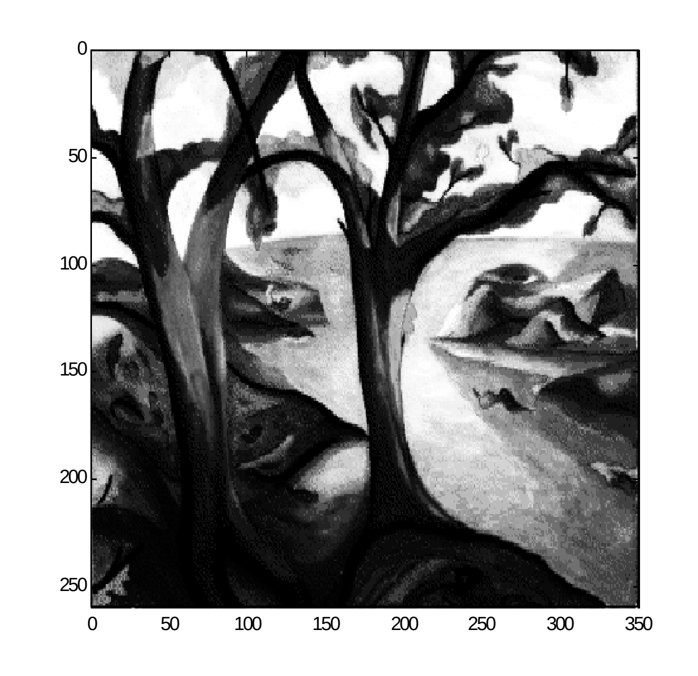

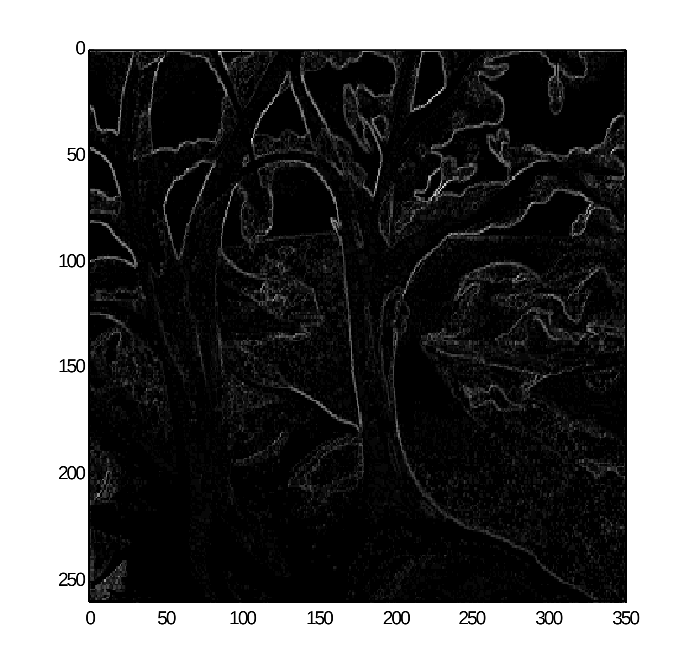

Wavelet Based Methods in Image Processing

Lecture 2.a - Fourier Transform

Applied Mathematics Seminar

S. Allen Broughton

Lecture 2.a

- signal samples and time domain
- model wave forms and frequency domain
- signal analysis & synthesis
- Discrete Fourier transform - DFT
- extension to 2D
- convolution theorem
- restoration

Signal Samples and Time Domain

- $X = [X(0),\ldots,X(N-1)]$ vector of length N
- $X(i) = f(i/N)$: sample analog signal f
- $X(i+N) = X(i)$: periodic extension (mod N)
- time domain $\{0,1/N,\ldots,(N-1)/N\}$ or $\{0,1,\ldots,N-1\}$

Model Wave Forms

- model wave forms

$$E_k(n) = \exp(2\pi i(k \cdot \frac{n}{N}))$$

- frequency domain

$$k = 0, \ldots, N - 1$$

- Maple script - waveform.mws

Signal Analysis & Synthesis - 1

- energy of a signal (* = conjugate transpose)

$$\mu(X, X) = X \cdot X = X^* X = \sum_{n=0}^{N-1} |X(n)|^2$$

- analysis of a signal -*"energy content at a frequency"*

$$\tilde{X}(r) = E_k \cdot X = E_k^* X$$

$$= \sum_{n=0}^{N-1} X(n) \exp(-2\pi i(k \cdot \frac{n}{N}))$$

Signal Analysis & Synthesis - 2

- synthesis or reconstruction for a signal

$$X = \sum_{k=0}^{N-1} a_k E_k$$

$$a_k = \frac{\tilde{X}(k)}{N}$$

Signal Analysis & Synthesis - 3

- additivity of energy

$$\mu(X, X) = \frac{1}{N}\sum_{k=0}^{N-1} |\tilde{X}(k)|^2 = N\sum_{k=0}^{N-1} |a_k|^2$$

Discrete Fourier Transform (DFT) - Matrix definition

- set $F = [E_0, E_1, \ldots, E_{N-1}]^*$

$$F(k,l) = \exp(-2\pi i \frac{kl}{N})$$

- then $\tilde{X} = F X$
- and $F^* F = \frac{1}{N} I$

- hence (inversion formula)

$$X = \frac{1}{N} F \tilde{X}$$

Discrete Fourier Transform Pictures

- Matlab scripts
  - dft1demo1.m, dft1demo2
  - dft2demo1.m, dft2demo2.m ,dftdemo3.m

Convolution Theorem

- Fourier transform takes convolution to pointwise multiplication

$$X * Y = \tilde{X} \bullet \tilde{Y}$$

- Eigenvector interpretation

$$X * E_k = H_X E_k = \tilde{X}(k) E_k = \lambda E_k$$

$$gain = |\lambda|, phaseshift = \angle \lambda$$

2D Discrete Fourier Transform

- Let X be an *mxn* image then

$$\tilde{X} = F_m X F_n^t = F_m X F_n$$

- 2D convolution theorem holds

$$X * Y = \tilde{X} \bullet \tilde{Y}$$

Simple Restoration -1

- Let X be an true image and Y the blurred and noisy image then

$$Y = M * X + E$$

- where M is a blurring mask and E is the noise
- ideally, find another mask R such that

$$R * M = id \qquad R * E = 0$$

- can't be done in practice but you try your best

Simple Restoration -2

$$Y \xrightarrow{F} \tilde{Y} \xrightarrow{process} \tilde{Z} \xrightarrow{F^{-1}} Z$$

- where $\tilde{Z} = \tilde{R} \bullet \tilde{M} \bullet \tilde{X} + \tilde{R} \bullet \tilde{E}$
- work in Fourier domain

$$R * M = id \rightarrow \tilde{R} \bullet \tilde{M} = id$$

- or $\tilde{R} = \tilde{M}^{-1}$ pointwise

Simple Restoration - 3

- Matlab Example - ansmid3.m

Issues in Restoration - 1

- Model issues
  – Good information on the blurring model *M* needs to be available, it can sometimes be found by experimentation
  – The distribution of *E* needs to be known
  – If *M* and *E* are unknown then you need to estimate them from the degraded image

Issues in Restoration - 2

- Computational issues
  - $\tilde{M}$ has zeros or small entries
  - thus $\tilde{R} = \tilde{M}^{-1}$ doesn't exist or is very large
  - and so $\tilde{R} \bullet \tilde{E}$ "noises up" the restored image