Image Restoration

EE 528 Digital Image Processing

- All images taken from Gonzalez and Woods online slides

http://www.imageprocessingplace.com/DIP/dip_faculty/classroom_presentations_downloads.htm

- Material mostly based on A.K. Jain’s book
- Some topics taken from Gonzalez Woods
- Intuition & images on slides, math details will be covered in class or is in the book

Image Enhancement or Restoration

- Most of what we learnt in Image Enhancement chapter can also be classified as Image Restoration techniques. Specifically
  - Linear filtering (low pass for noise reduction, high pass for edge sharpening, band-pass for both)
  - Median filtering (for salt and pepper noise),
  - Log-domain filtering and other nonlinear techniques

Inverse & Pseudo-inverse Filters

- Inverse Filter
  - Assumes no noise, only blurring.
  - Blurring filter known
  - In case of noise
    - If blurring filter has zeros at some frequencies (which it will since it is a low-pass filter), those frequencies will be amplified in the noise
- Pseudo-inverse filter:
    - removes the problem at zero (or near zero) frequencies, but still amplifies noise at other frequencies where the blurring filter response is not zero but small

Image blurred by atmospheric turbulence & with additive noise

a b
c d

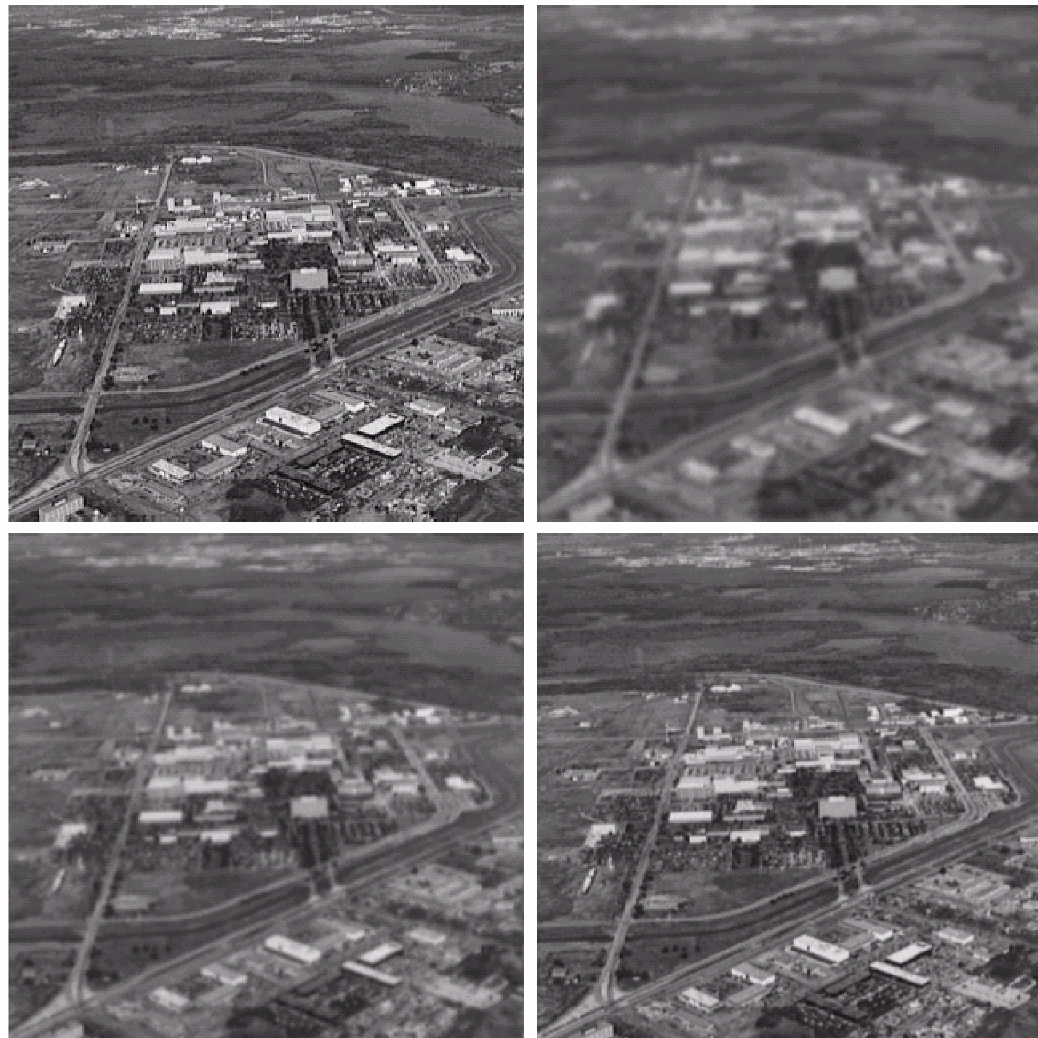


FIGURE 5.25 Illustration of the atmospheric turbulence model. (a) Negligible turbulence. (b) Severe turbulence, $k = 0.0025$. (c) Mild turbulence, $k = 0.001$. (d) Low turbulence, $k = 0.00025$. (Original image courtesy of NASA.)

$$H(u, v) = e^{-c(u^2+v^2)^{\frac{5}{6}}}$$

Inverse v/s Pseudo-inverse filtering

a b
c d

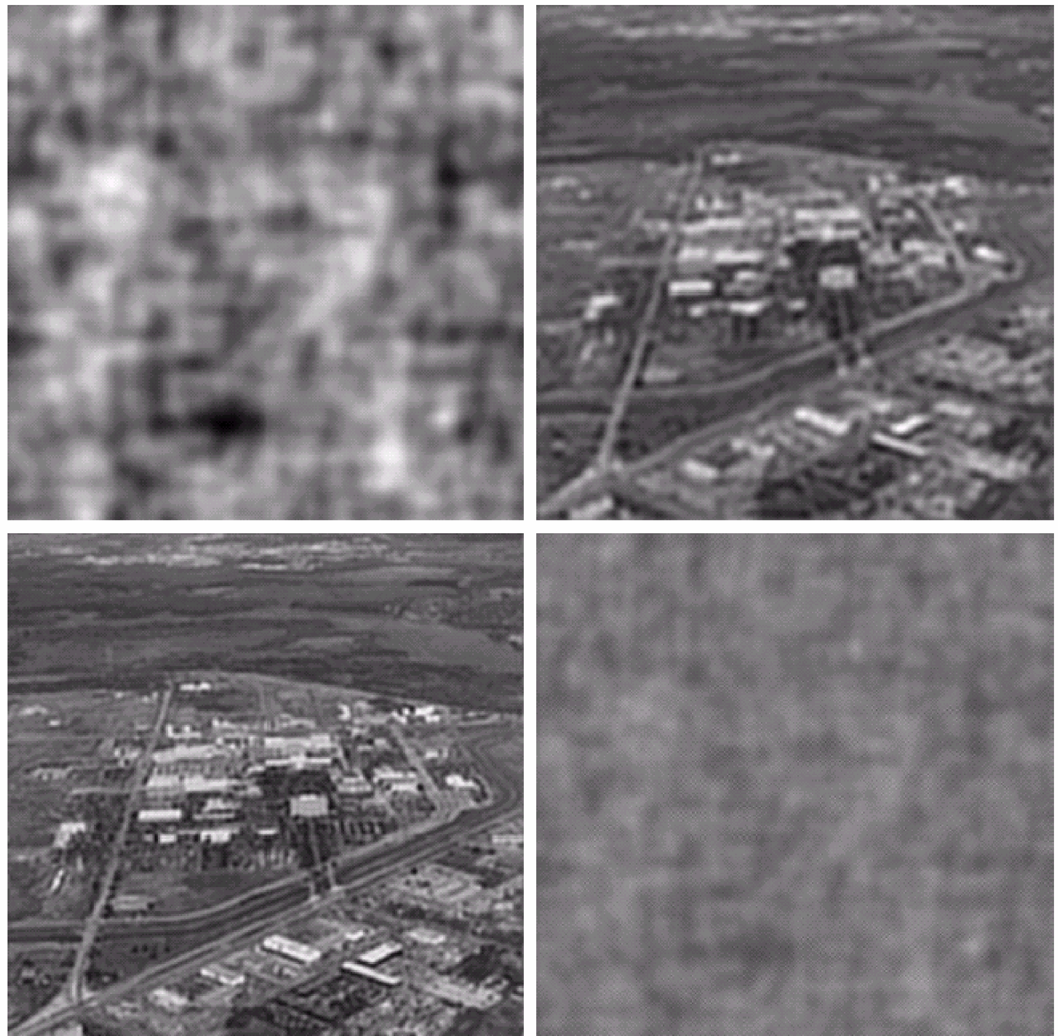


FIGURE 5.27 Restoring Fig. 5.25(b) with Eq. (5.7-1). (a) Result of using the full filter. (b) Result with H cut off outside a radius of 40; (c) outside a radius of 70; and (d) outside a radius of 85.

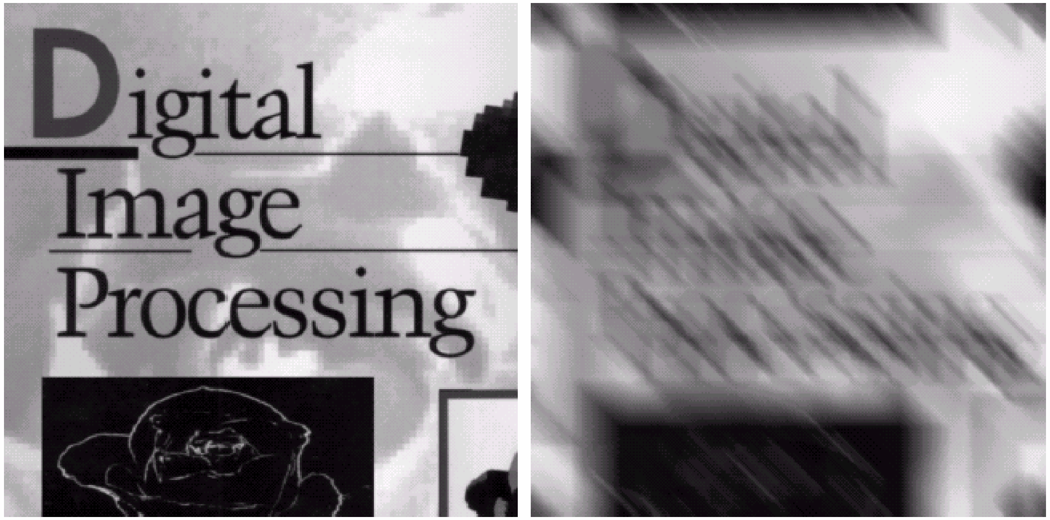
Wiener Smoother

- Assumes image is blurred and has additive noise (independent of image)
- Need to know
  - Blurring filter
  - Noise covariance
  - True image autocorrelation
  - Mean of noise & of true image (or assume zero mean)

- Gives "linear MMSE" estimate: linear filter with least expected value of MSE w.r.t. the true image
- Truly MMSE when the observed and true image are jointly Gaussian

Motion blurred image

$$H(u,v) = \frac{\sin(\pi V T u)}{\pi V u}$$



a b

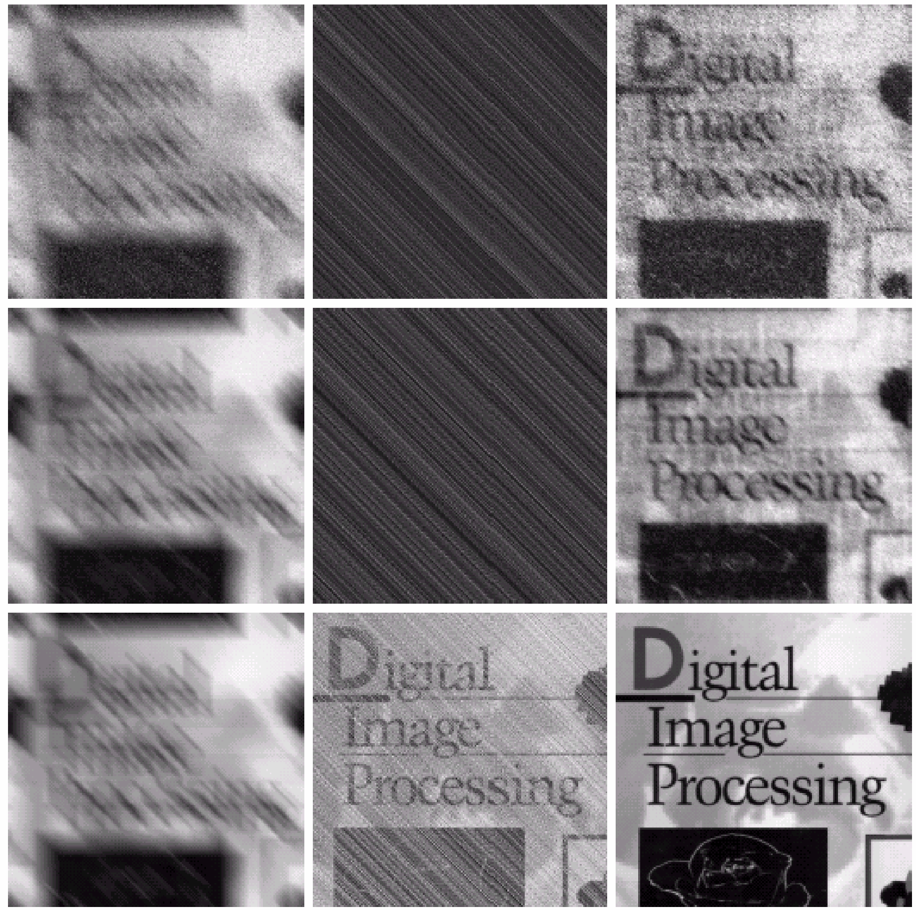
FIGURE 5.26 (a) Original image. (b) Result of blurring using the function in Eq. (5.6-11) with $a = b = 0.1$ and $T = 1$.

Pseudo-inverse v/s Wiener

Column 2: Pseudo-inverse

Column 3: Wiener

Maximum noise in row 1, least in row 3



a b c
d e f
g h i

FIGURE 5.29 (a) Image corrupted by motion blur and additive noise. (b) Result of inverse filtering. (c) Result of Wiener filtering. (d)–(f) Same sequence, but with noise variance one order of magnitude less. (g)–(i) Same sequence, but noise variance reduced by five orders of magnitude from (a). Note in (h) how the deblurred image is quite visible through a "curtain" of noise.

Wiener smoother

- Observed: v. True: u, convolution: *
- Can use orthogonality argument to show that the MMSE restored image u_{hat} = g * v, where g (Wiener smoother) satisfies
  - $g(k,l)*r_{vv}(k,l) = r_{uv}(k,l)$ for all (k,l), or equivalently
  - $G(f_1,f_2) = S_{uv}(f_1,f_2)\, S_{vv}^{-1}(f_1,f_2)$
- Assume v = u * h + n, n: noise
- Need to know r_{uu}, h, r_{nn}. h: blurring filter
- Compute $r_{uv} = h(-k,-l) * r_{uu}(k,l)$
- Compute $r_{vv} = h(k,l) * r_{uv} + r_{nn}$

Properties of Wiener Smoother

- Non-causal: okay for image processing
- For time series applications: need to find the best causal filter that minimizes expected MSE: more complicated: **Wiener filter**

- Wiener computes correlations etc assuming all signals are zero mean
- If not, then subtract out the means first and then compute auto-correlations (in other words, always using auto-covariances)

- Output noise is NOT white.
- 2D Wiener: not separable even if h, r_{uu} are

FIR Wiener

- Exact Wiener filter or smoother are infinite impulse response (IIR)
- IIR: expensive to implement
- Many IIR coefficients are small, one solution is to truncate it. Or
- Design an FIR Wiener
  - Find the best (least expected MSE) filter with (2M+1) x (2M+1) taps
  - This will give lower MSE than just truncating the IIR filter

FIR Wiener Algorithm

- Assume noise is white and assume noise power known, i.e. σ_n^2 known
- Estimate r_{vv} from the observed image
- Solve for $r_{uv}(k,l)$ from
  - $r_{vv}(k,l) = r_{uv}(k,l) * h(k,l) + \sigma_n^2 \delta(k,l)$
  - Assume r_{uv} zero for more than a few taps
  - Need $r_{uv}(k,l)$ only for $-M <= k,l <= M$
- g satisfies

$$r_{vv}(k,l) * g(k,l) = r_{uv}(k,l), \quad -M <= k,l <= M$$

FIR Wiener – No blurring

- h = identity
- Only need to know SNR σ_u^2 / σ_n^2. Say SNR = a
  - $\sigma_n^2 = r_{vv}(0,0) / (1+a)$
  - $r_{uu}(0,0) = a\, r_{vv}(0,0) / (1+a)$
  - $r_{uu}(k,l) = r_{vv}(k,l)$ if $k \neq 0$ or $l \neq 0$
  - $r_{uv}(k,l) = r_{uu}(k,l)$
- Compute different Wiener filters for diff image blocks
- Choose different length Wieners, e.g. choose M s.t. output SNR σ_u^2 / σ_e^2 is roughly constant

- Need a longer tap Wiener if more noise or more blurring.
- If zero noise, Wiener approaches inverse filter
- Summary
  - Wiener smoother
  - Wiener filter or causal Wiener (mostly needed for 1D)
  - FIR Wiener (causal or non-causal)
  - Computing r_{uu}
  - Can also use AR model to get r_{uu} , i.e. use a clean image to estimate an AR model for the image: that can be used to compute r_{uu}

Smoothing splines

- Main idea: find the “smoothest” function that “fits the data (observed image)”, i.e. error between the observed image and smooth function is below a threshold
- “Smoothness” quantified in various ways, one way is to minimize roughness, i.e. find the function that “fits the data” and has the lowest sum of double derivatives
- This is called a smoothing spline

Constrained Least Squares

- Find a maximally “smooth” restored image, i.e. find u_{hat} so that

$J = || q(m,n) * u_{hat}(m,n)||$ is minimized and

$||v(m,n) – h(m,n)*u_{hat}(m,n)||^2 <= \varepsilon^2$ #

- Solution:

$U_{hat} = H^* V / (|H|^2 + \gamma|Q|^2)$

γ computed s.t. # satisfied with equality

- This is the Wiener smoother when $S_{nn} = \gamma$ and $S_{uu} = 1/|Q|^2$

- Usually choose q to be the discrete Laplacian
- Smoothing splines is same as constrained LS when h = identity, i.e. no blurring

Constrained LS solution

a b c

FIGURE 5.30 Results of constrained least squares filtering. Compare (a), (b), and (c) with the Wiener filtering results in Figs. 5.29(c), (f), and (i), respectively.

Geometric distortion

- Also called “rubber-sheet” transformations
- Image pixel locations distorted, e.g. translate the image or rotate it or affine deform the image (due to camera motion)
- Consider distorted image v generated by

$v(x',y') = u(x,y)$, $x' = r(x,y)$, $y' = s(x,y)$

$r(x,y) = c1\ x + c2\ y + c3\ xy + c4$

$s(x,y) = c5\ x + c6\ y + c7\ xy + c8$

Restoring geometric distortions

- Step 1: estimate the distortion, i.e. find a set of 4 (or more) corresponding points in the two images and compute the coefficients c1 to c8
- Step 2:
  - $u_{hat}(x,y) = g(r(x,y), s(x,y))$
  - r(x,y), s(x,y) may not be integers
  - Need grey scale interpolation
    - Zeroth order hold: g(round(r(x,y)), round(s(x,y)))
    - Bilinear: use floor and ceil of r(x,y), s(x,y) to compute interpolating function parameters

Corresponding points

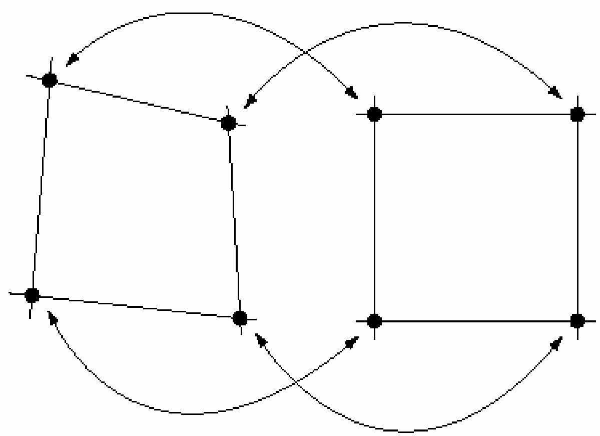


FIGURE 5.32 Corresponding tiepoints in two image segments.

Geometric distortion/restoration

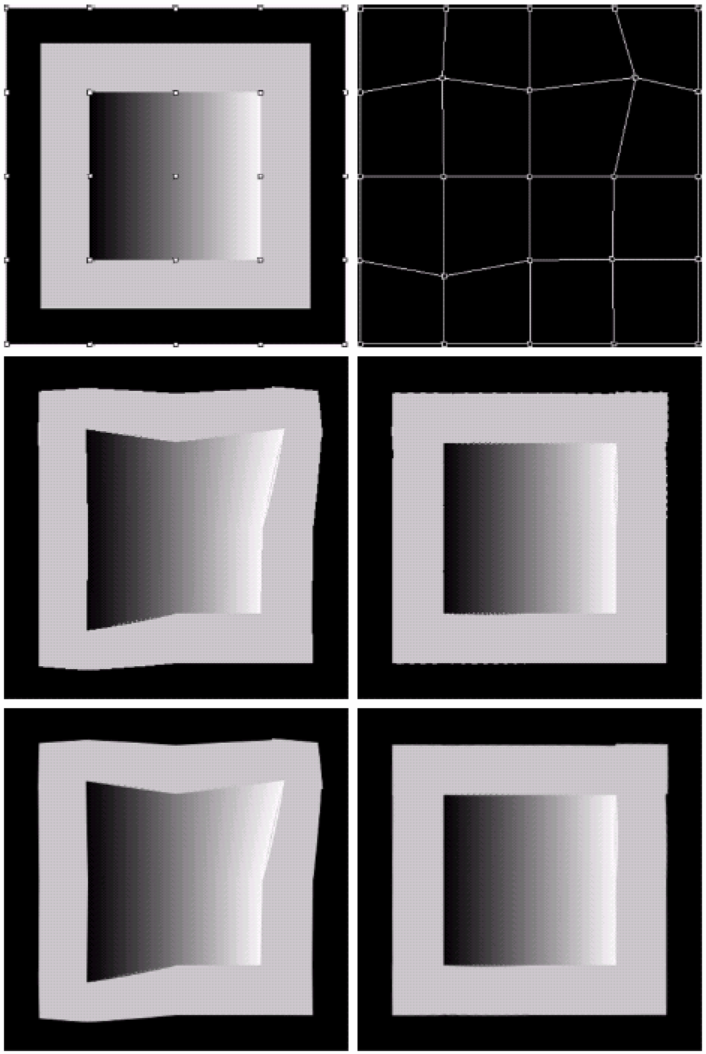


FIGURE 5.34 (a) Image showing tiepoints. (b) Tiepoints after geometric distortion. (c) Geometrically distorted image, using nearest neighbor interpolation. (d) Restored result. (e) Image distorted using bilinear interpolation. (f) Restored image.

Next class

Important difference from other restoration methods

- To estimate the geometric distortion, i.e. c1 through c8, need at least 4 corresponding point locations in the true image and the distorted image
  - Correcting for geometric distortion is actually an Image Registration technique: assumes both images given

- Other restoration techniques only use the observed image and “some” other information (e.g. autocorrelation of the true image or at least knowledge of the blurring filter)

More on Geometric distortions

- Finding corresponding points is the difficult problem
- Commonly used techniques
  - Feature matching
    - Color, KLT tracker, local histogram, local PCA, texture
  - Corner detection methods
  - Curvature vertices (maxima/min/discontinuities)
  - Find the geometric distortion between two whole contours
  - Jointly register & segment

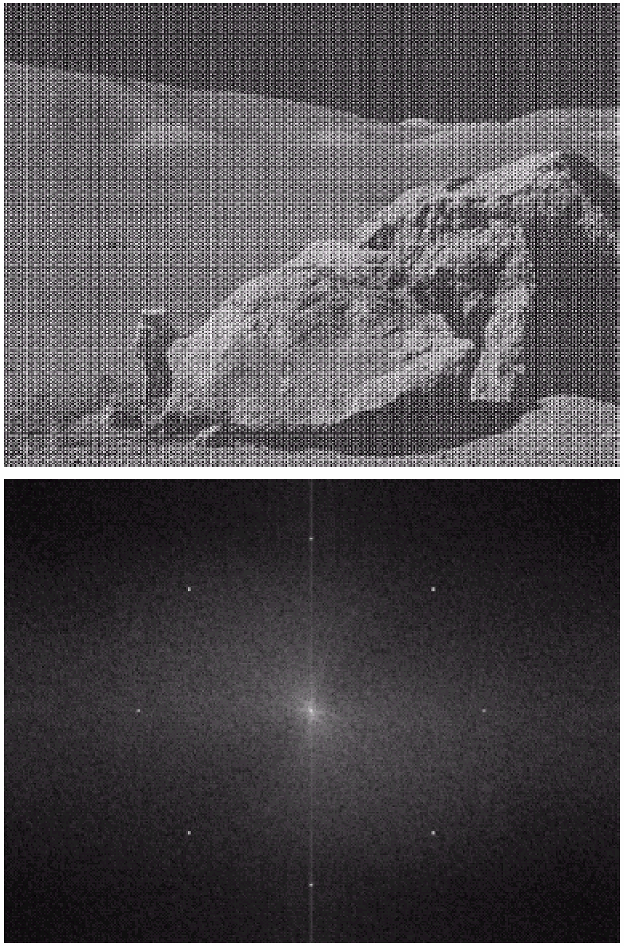
Registering 2 contours

- Contours may have different lengths (if there has been a size change). Solution:
  - Sub-sample, uniformly along arclength, both contours to a fixed number of points M. Use these points as "corresponding points"
    - Computing arclength: in class
    - To do the above robustly: B-spline control points
  - Easier but approx solution: assume the points you get are already uniformly sampled, just resample to a fixed number of points M
    - Details in class

Notch filter

- Filter out a certain frequency or a certain small band of frequencies
- All called “band-reject” filter
- Easiest to implement in the frequency domain

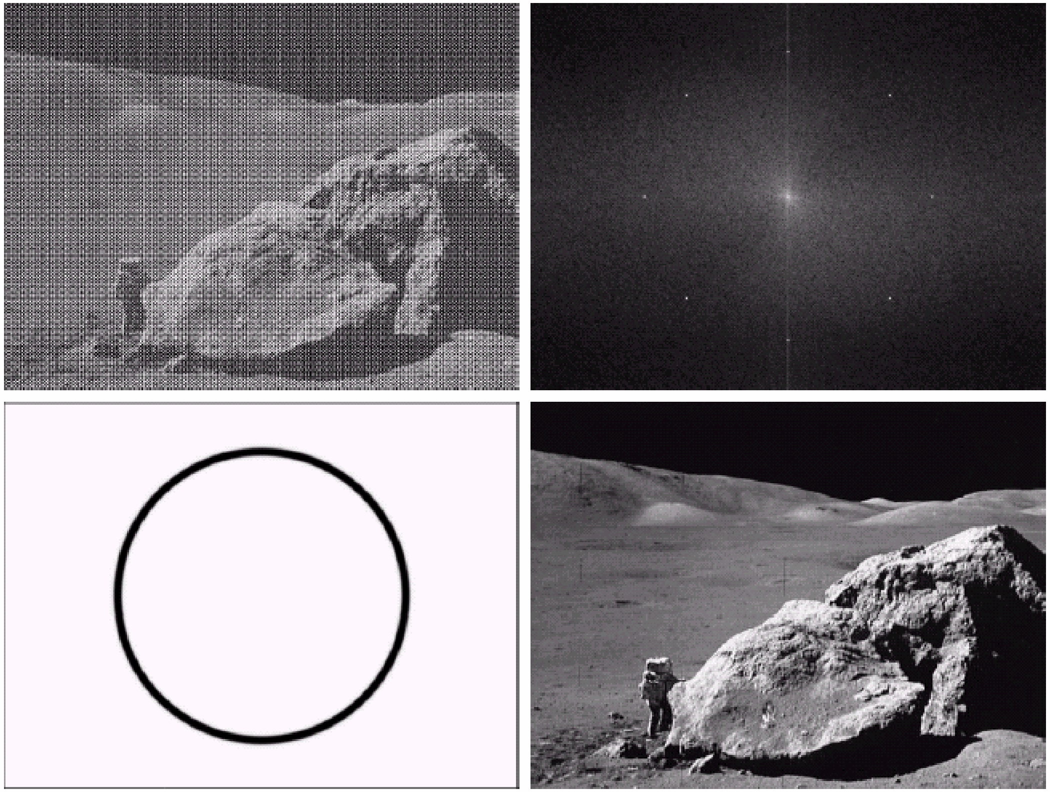
Image corrupted by sinusoidal noise



a
b

FIGURE 5.5 (a) Image corrupted by sinusoidal noise. (b) Spectrum (each pair of conjugate impulses corresponds to one sine wave). (Original image courtesy of NASA.)

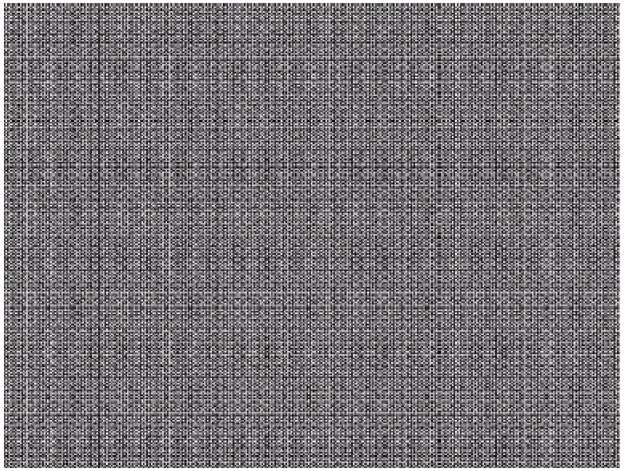
Notch filtering/Band-reject filtering



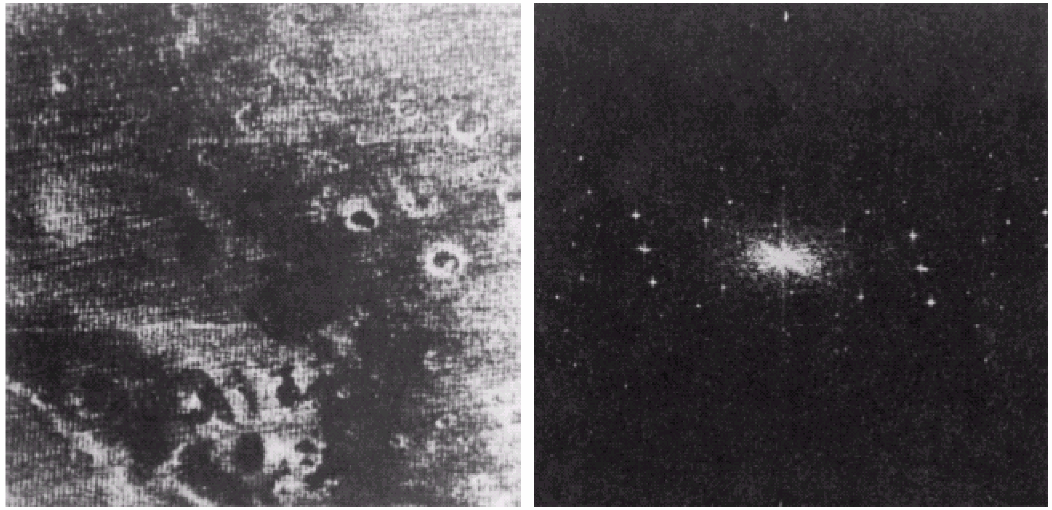
a b
c d

FIGURE 5.16
(a) Image corrupted by sinusoidal noise. (b) Spectrum of (a). (c) Butterworth bandreject filter (white represents 1). (d) Result of filtering. (Original image courtesy of NASA.)

FIGURE 5.17 Noise pattern of the image in Fig. 5.16(a) obtained by bandpass filtering.



Another example



a b

FIGURE 5.20 (a) Image of the Martian terrain taken by *Mariner 6.* (b) Fourier spectrum showing periodic interference. (Courtesy of NASA.)

FIGURE 5.23 Processed image. (Courtesy of NASA.)

Speckle noise

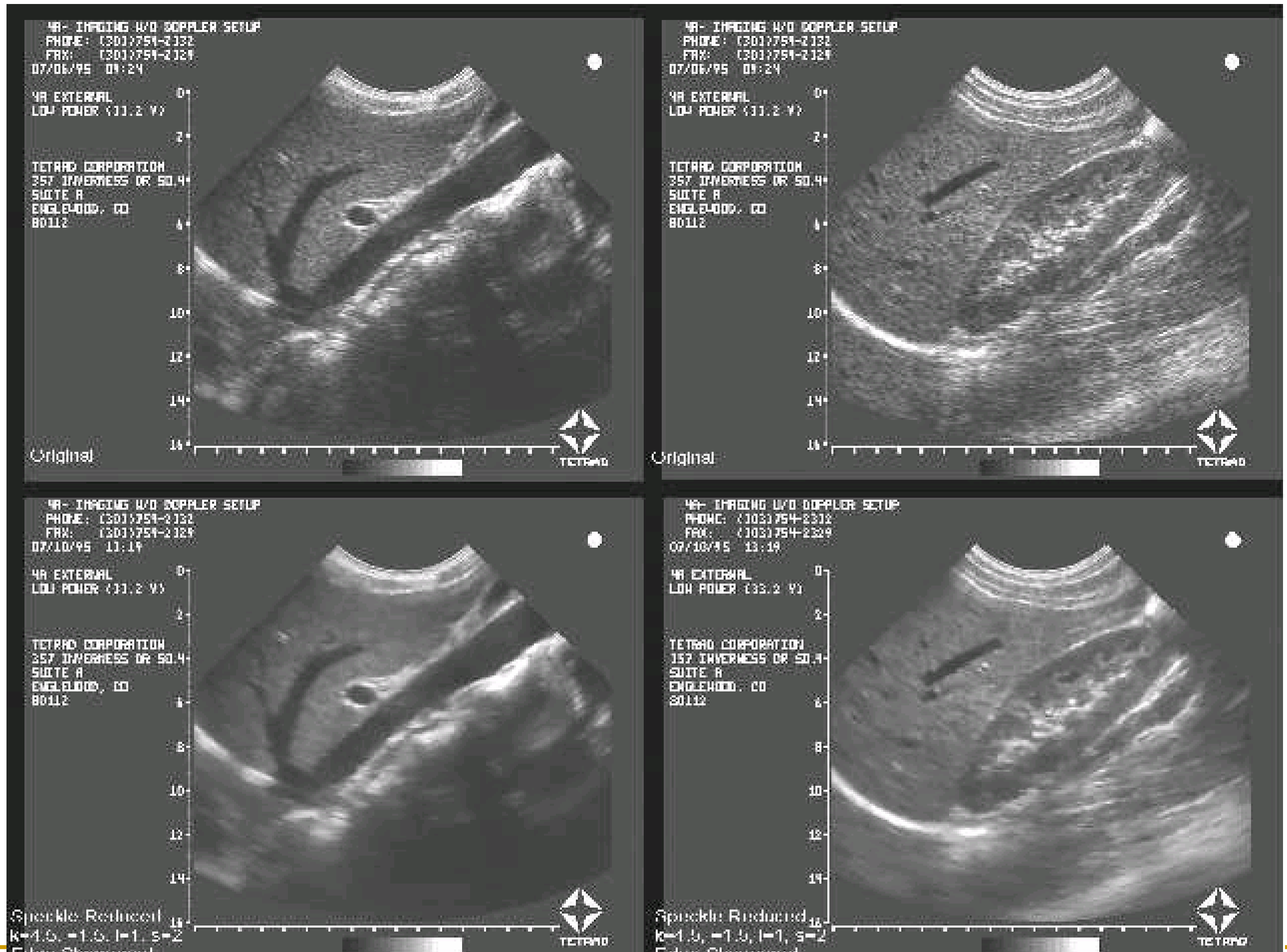
- Occurs in all coherent imaging systems
- Examples
  - Ultrasound
  - SAR (Synthetic Aperture Radar)
- Effect of interference of energy from randomly distributed scatters, too small to be resolved by the imaging system
- Occurs when object roughness is of the order of the incident radiation's wavelength

Speckle noise model

- Model as infinite sum of i.i.d. phasors with random amplitude and phase, multiplied with the image
  - Details in class
- If low resolution image, an approximate model is
  - $v(x,y) = u(x,y)\, s(x,y) + n(x,y)$
  - n = additive detector noise (already discussed)
  - s = speckle noise
  - s(x,y) = sum of squared magnitude of iid phasors
  - s(x,y): iid exponential with parameter $1/(2\sigma_a^2)$
  - Details in class
  - Similar to “multi-path fading” in communication channels

Ultrasound speckle images

- http://www.ljbdev.com/speckle.html



Speckle noise reduction

- Simplest: if multiple ultrasound images available: average them
  - If the multiple images are not "registered" (geometric distortion b/w them), first register as studied in last class, then average
  - Effect of averaging: the speckle will be more constant across pixels, but will NOT go away
- Techniques for multiplicative noise
  - **Homomorphic filtering**
  - Take logarithm: log v(x,y). log s(x,y) is then additive noise. Assume it to be Gaussian and apply a Wiener filter to log v(x,y)
  - Assumption holds strictly only if s(x,y) were log-normal which it is not. But Wiener still works!

Another application of Homomorphic filtering

- Any image consists of an illumination component and a reflectance component
  - $I(x,y) = i(x,y)\, r(x,y)$
  - Illumination: i, Reflectance: r
- We are interested in reflectance only: which is due to object texture
- Illumination will be non-uniform if light falling at an angle. Want to get rid of it.
- Take log of image, take DFT, suppress low frequencies (assumes illumination is low freq), take I-DFT, and exp

Bayesian methods for Restoration

- General idea:
  - Given p(v|u) (data term or likelihood) & p(u) (prior)
  - Estimate maximizer of p(u|v) : called the MAP solution
- Wiener is MAP when u and v are jointly Gaussian
  - Weiner gives E[u|v] = conditional mean = MMSE estimate
  - MAP is $\max_u p(u|v)$
  - Since p(u|v) is Gaussian, MAP = conditional mean
- Constrained LS or actually Regularized LS
  - Min $||q^*u||^2 + ||v - H u||_{Rn}^2$
  - Solution is Bayesian with a smoothness prior q
- Nonlinear MAP: v = f(Hu) + n, prior on u given

Blind De-convolution: one solution

- Computing $H(f_1,f_2)$
  - Compute $S_{vv}(f_1,f_2)$ from observed image
  - Assume $S_{uu}(f_1,f_2)$ known (FT of r_{uu}) from training data
  - Assume $S_{nn}(f_1,f_2) = \sigma_n^2$ is known

  - $\log |H|^2 = \log (S_{vv} - S_{nn}) - \log S_{uu}$
- This does not give the phase of H: in many cases H is linear phase (e.g. due to motion blur) or zero phase. If need phase: compute S_{uv} to get it
- Use H in the Wiener filter

Models for the true image

- Wiener filter requires r_{uu}
  - It may be estimated from r_{vv}, but there you are assuming the noise covariance is known and h (blurring function) is known
- When h unknown, definitely need to know r_{uu} to estimate h

- One solution to obtain r_{uu}: assume you have at least one true image available as “training data”
  - Model the image as an MRF (Markov Random Field)
  - Usually suffices to use the 4 nearest neighbors for prediction
  - Details in class or on pages 206 -208 of AK Jain.

More on Blind De-convolution