Deconvolution

Frédo Durand
MIT EECS 6.815/6.865
Aberrations result in blur

- E.g.
  - Spherical aberration, Diffraction, Camera shake

- Is it shift invariant (convolution)?
  - Not quite. For example, we typically get spherical aberrations in the center of the image but comatic aberrations in the periphery
    - But locally yes.

- In what follows, we use a shift-invariant model

- And we assume we know the blurring kernel
  - e.g. we have calibrated the lens
  - see DXO http://www.dxo.com/
Questions?
Recall Blurring

original

Blurred

0.3

0
Can we undo blur?

Not easy, even when we know the kernel
How can we deblur?

- Play with Photoshop unsharp mask
  - unlikely to cancel an arbitrary blur
- Write down the linear equation:
  blurred image = linear combination of sharp image
  - And invert
- Can we make this inversion problem simpler?
- Yes! Diagonalize the blur matrix
  - Fourier to the rescue!
  - At least to analyze the problem
Recall convolution theorem

- Convolution in space is a multiplication in Fourier
- Note $y$ the observed blurry image and $x$ the original sharp one
- $y = g \ast x$ in the spatial domain
- $Y = GX$ in the Fourier domain
  - Each component does not depend on the other ones
Invert the convolution theorem

- Given \( y = g \otimes x \) and \( g \), we seek an estimate \( x' \) of \( x \)
- How do you invert a multiplication?
  - Division!
- \( X'(\omega) = Y(\omega)/G(\omega) \)

DECONVOLUTION IS A DIVISION IN THE FOURIER DOMAIN!

- Which means it is also a convolution in the spatial domain, by the inverse Fourier transform of \( 1/G \)
Questions?

- Given $y = g \otimes x$ and $g$, we seek an estimate $x'$ of $x$
- How do you invert a multiplication?
  - Division!
- $X'(\omega) = Y(\omega) / G(\omega)$

- DECONVOLUTION IS A DIVISION IN THE FOURIER DOMAIN!
- Which means it is also a convolution in the spatial domain, by the inverse Fourier transform of $1/G$
Potential problem?

- Deconvolution is a division in the Fourier domain
- Division by zero is bad!
  - Information is lost at the zeros of the kernel spectrum \( G \)
Noise problem

- Even when there is no zero, noise is a big problem
- If G has small number, division amplifies noise
- if \( y = g \otimes x + v \) where \( v \) is additive noise
- \( Y = GX + V \)
- \( X' = \frac{(GX + V)}{G} = X + \frac{V}{G} \)
- \( V \) is amplified by \( \frac{1}{G} \). This is why you typically get more high-frequency noise with deconvolution
Questions?
Tradeoff

- Can we tweak $1/G$ to reduce output noise?
- Maybe if we use something smaller than $1/G$
  - we won't amplify noise as much
  - but the inversion won't be as correct

- For this, let's assume that we know the expected noise power spectrum
Tradeoff: example for one $\omega$

- $X(\omega) = 10; \ G(\omega) = 0.1; \ Y(\omega) = 1 + V(\omega)$

- Try $1/G = 10$
  - if $V = 5$: $Y = 6; \ X' = 60 \Rightarrow \varepsilon = 50$
  - if $V = 0$: $Y = 1; \ X' = 10 \Rightarrow \varepsilon = 0$
  - if $V = -5$: $Y = -4; \ X' = -40 \Rightarrow \varepsilon = -50$

- Try $H = 5$ instead
  - if $V = 5$: $Y = 6; \ X' = 30 \Rightarrow \varepsilon = 20$
  - if $V = 0$: $Y = 1; \ X' = 5 \Rightarrow \varepsilon = 5$
  - if $V = -5$: $Y = -4; \ X' = -20 \Rightarrow \varepsilon = -30$

- better on average
Wiener filtering

- Find the gain $H$ that minimize $\|X'(\omega) - X(\omega)\|^2$
  where $X' = HY$
- We need to know the signal noise ratio $X/N$
- Optimal filter
  \[
  \frac{1}{G(\omega)} \left[ \frac{|G(\omega)|^2}{|G(\omega)|^2 + 1/SNR(\omega)} \right]
  \]
- See derivation at
  http://en.wikipedia.org/wiki/Wiener_deconvolution
  • careful, their notations are different from mine
Wiener filtering

- hand-wavy version: divide by $1/(G+N)$

- See derivation at http://en.wikipedia.org/wiki/Wiener_deconvolution
  - careful, their notation is different from mine
Questions?
Richardson Lucy

- Based on Poisson noise assumption
  - Noise amount depends on value
- Iterative (EM)
  - Estimate values given noise
  - Deduce noise given values

http://people.csail.mit.edu/dgreensp/may/deconvlucy.html
Questions?
Blind deconvolution

- So far we have assumed we know the kernel
- When both $x$ and $g$ are unknown, the problem is badly ill-posed
- It is called blind deconvolution