Image Deblurring Primer

CS635 Spring 2017

Daniel G. Aliaga
Department of Computer Science
Purdue University
Image Blurring

• Where does blur come from?
  – Optical blur: camera is out-of-focus
  – Motion blur: camera or object is moving

• Why do we need deblurring?
  – Visually annoying
  – Wrong target for compression
  – Bad for analysis
  – Numerous applications
Optical Example

Before

After
Example

Observed image of Saturn

Restored image
Example

![Example image with car before and after processing]
Modeling Blurring Process

\[ x(m,n) \xrightarrow{h(m,n)} y(m,n) \] (input) (output)

\[ h(m,n) = \text{blurring filter (or kernel, or PSF)} \]

\[ w(m,n) \sim N(0, \sigma_w^2) = \text{additive white Gaussian noise} \]
Classic Example

\[ x(m,n) \rightarrow (\text{horizontal blur}) \rightarrow y(m,n) \]
Types of “Deblurring”

• Blind Deblurring
  – Blurring kernel unknown

• Non-blind Deblurring
  – Blurring kernel known

• Bounded vs Nonbounded Deblurring
  – If bounded, pixel values are kept in a fixed range
    (less studied)
Gaussian filter can be used to approximate out-of-focus blur

\[ h(m, n) = \exp\left( -\frac{m^2 + n^2}{2\sigma^2} \right) \]

\[ H (w_1, w_2) = \exp\left( -\frac{w_1^2 + w_2^2}{2\sigma^2} \right) \]
Blurring Filter Example

Motion blurring along a line

MATLAB code: 
\[ h = \text{FSPECIAL('motion',9,30)}; \]
Inverse Filter

To compensate the blurring, we seek

\[ h_{\text{combi}}(m, n) = h(m, n) \otimes h^I(m, n) = \]

\[ \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} h^I(m - k, n - l)h(k, l) = \delta(m, n), \forall (m, n) \]

\[ H^I(w_1, w_2) = \frac{1}{H(w_1, w_2)} \]
Problem: Zeros

\[ OTF = \frac{H(w_1, w_2)}{H(0,0)} \]

\[ MTF = |OTF| \]

The kernel/filter cancels some frequencies and causes “zeroes” which is bad – cannot recover
But It Can Work...

- Some approaches:
  - Inverse Filter
  - Wiener Filter
  - Lucy Richardson
  - And more!