Deblurring & Deconvolution

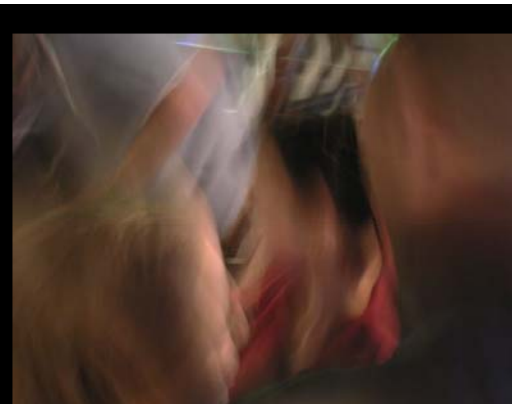
Lecture 10

Admin

- Assignment 3 due
- Last lecture
  - Move to Friday?
- Projects
  - Come and see me

Different types of blur

- Camera shake
  - User moving hands
- Scene motion
  - Objects in the scene moving
- Defocus blur [NEXT WEEK]
  - Depth of field effects



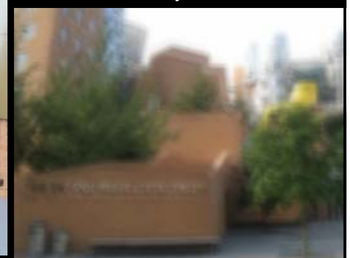
Overview

- Removing Camera Shake
  - Non-blind
  - Blind
- Removing Motion Blur
  - Non-blind
  - Blind

- Focus on software approaches

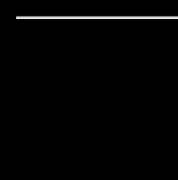
Let's take a photo

Blurry result



Slow-motion replay

Slow-motion replay



Motion of camera

Image formation model: Convolution

Blurry image

Input to algorithm

Sharp image

Desired output

Blur kernel

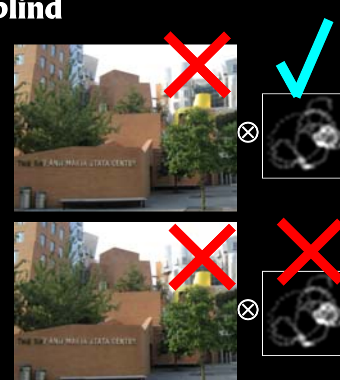
Convolution operator

Model is approximation

Assume static scene

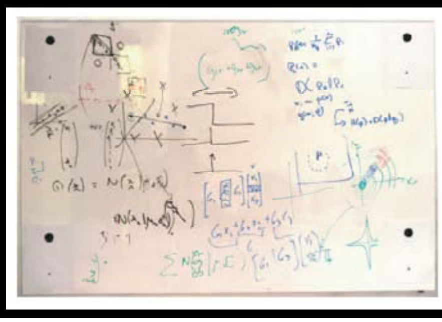
Blind vs Non-blind

- Non-blind
- Blind

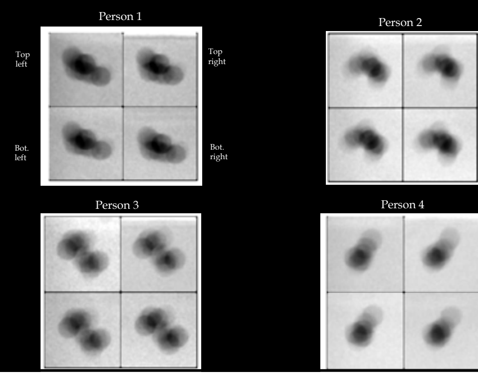


Camera Shake – is it a convolution?

8 different people, handholding camera, using 1 second exposure



Dots from each corner



What if scene not static?

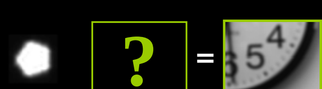
- Partition the image into regions

Overview

- Removing Camera Shake
  - Non-blind
  - Blind
- Removing Motion Blur
  - Non-blind
  - Blind

Deconvolution is ill posed

$$f \otimes x = y$$

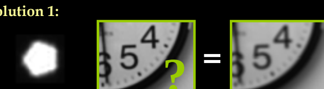


Slide from Anat Levin

Deconvolution is ill posed

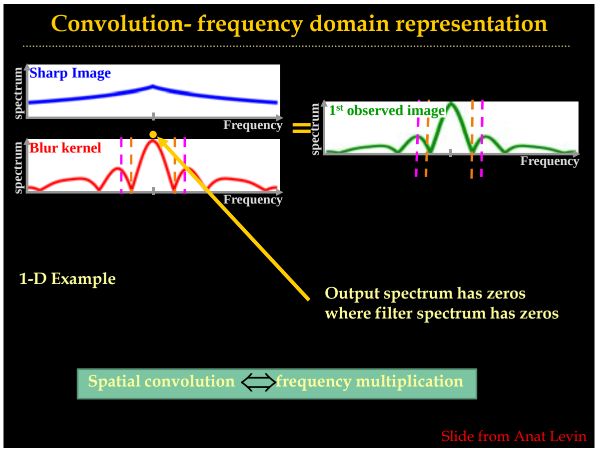
$$f \otimes x = y$$

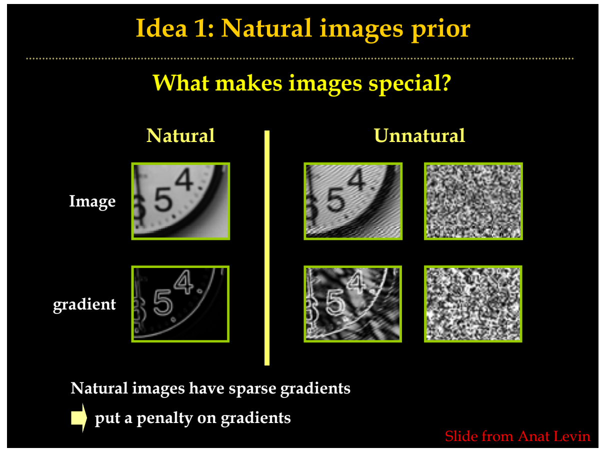
Solution 1:

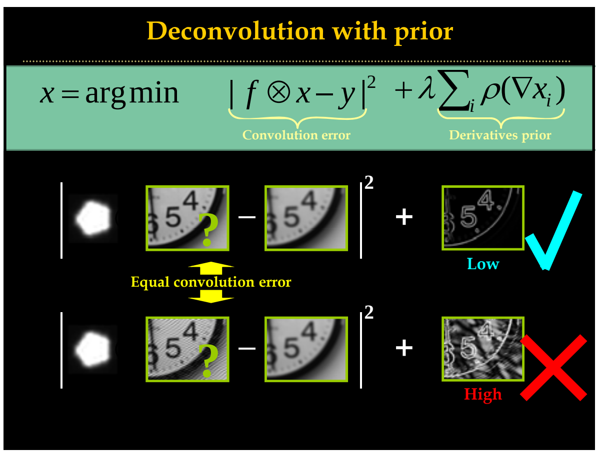


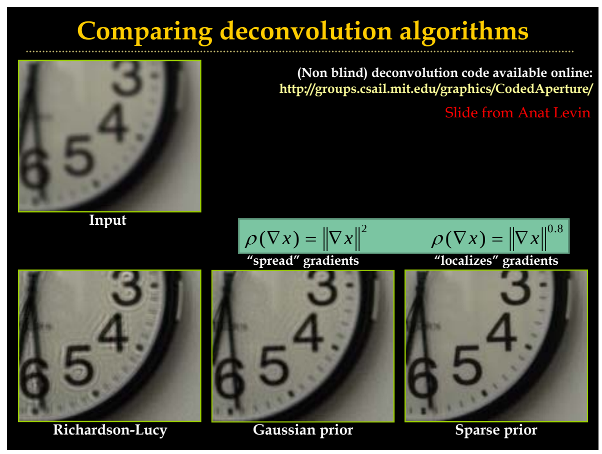
Solution 2:

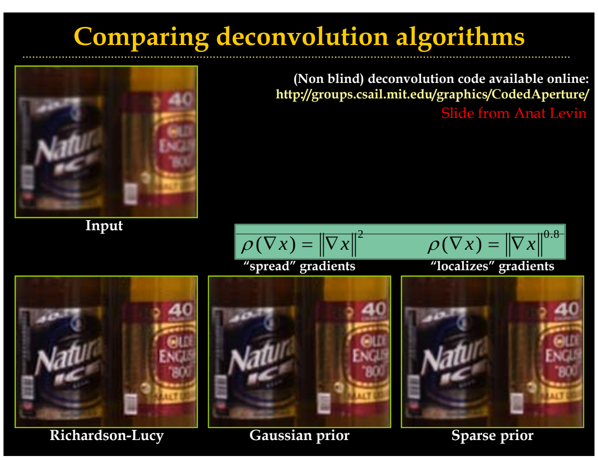
Slide from Anat Levin

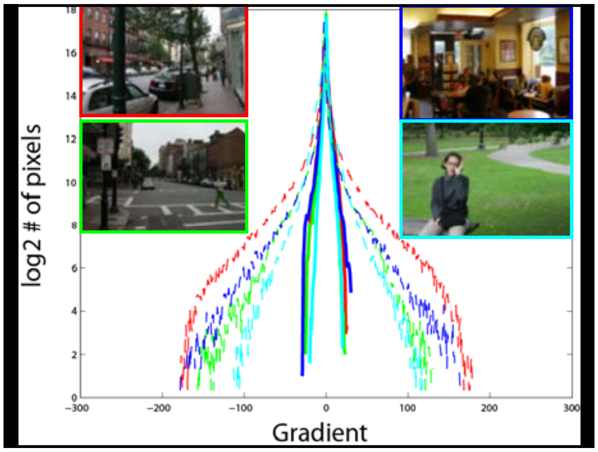

Convolution- frequency domain representation
Sharp Image
spectrum
Frequency
Blur kernel
spectrum
Frequency
=
1st observed image
spectrum
Frequency
1-D Example
Output spectrum has zeros
where filter spectrum has zeros
Spatial convolution ⟺ frequency multiplication
Slide from Anat Levin


Idea 1: Natural images prior
What makes images special?
Natural
Unnatural
Image
gradient
Natural images have sparse gradients
put a penalty on gradients
Slide from Anat Levin


Deconvolution with prior
$x = \arg\min \underbrace{| f \otimes x - y |^2}_{\text{Convolution error}} + \underbrace{\lambda \sum_i \rho(\nabla x_i)}_{\text{Derivatives prior}}$
Equal convolution error
Low
High


Comparing deconvolution algorithms
(Non blind) deconvolution code available online:
http://groups.csail.mit.edu/graphics/CodedAperture/
Slide from Anat Levin
Input
$\rho(\nabla x) = \|\nabla x\|^2$
"spread" gradients
$\rho(\nabla x) = \|\nabla x\|^{0.8}$
"localizes" gradients
Richardson-Lucy
Gaussian prior
Sparse prior


Comparing deconvolution algorithms
(Non blind) deconvolution code available online:
http://groups.csail.mit.edu/graphics/CodedAperture/
Slide from Anat Levin
Input
$\rho(\nabla x) = \|\nabla x\|^2$
"spread" gradients
$\rho(\nabla x) = \|\nabla x\|^{0.8}$
"localizes" gradients
Richardson-Lucy
Gaussian prior
Sparse prior



Application: Hubble Space Telescope

- Launched with flawed mirror
- Initially used deconvolution to correct images before corrective optics installed

Image of star

Non-Blind Deconvolution Matlab Demo

- http://groups.csail.mit.edu/graphics/CodedAperture/DeconvolutionCode.html

Overview

- Removing Camera Shake
  - Non-blind
  - Blind
- Removing Motion Blur
  - Non-blind
  - Blind

Removing Camera Shake from a Single Photograph

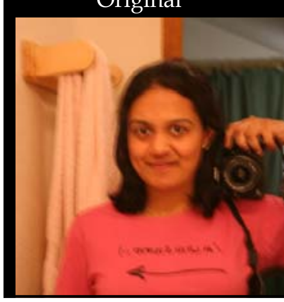
Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T. Roweis and William T. Freeman

Massachusetts Institute of Technology
and
University of Toronto

Overview

Joint work with B. Singh, A. Hertzmann, S.T. Roweis & W.T. Freeman

Original

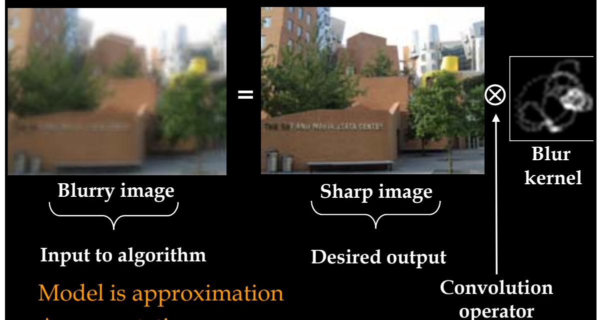


Our algorithm

Close-up

Original | Naïve sharpening | Our algorithm

Image formation process



Model is approximation

Assume static scene

Existing work on image deblurring

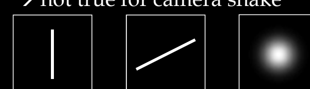
Old problem:

- Trott, T., "**The Effect of Motion of Resolution**", *Photogrammetric Engineering*, Vol. 26, pp. 819-827, 1960.
- Slepian, D., "**Restoration of Photographs Blurred by Image Motion**", *Bell System Tech.*, Vol. 46, No. 10, pp. 2353-2362, 1967.

Existing work on image deblurring

Software algorithms for natural images

- Many require multiple images
- Mainly Fourier and/or Wavelet based
- Strong assumptions about blur
  → not true for camera shake



Assumed forms of blur kernels

- Image constraints are frequency-domain power-laws

Existing work on image deblurring

Hardware approaches

Image stabilizers | **Dual cameras** | **Coded shutter**

Ben-Ezra & Nayar CVPR 2004 | Raskar et al. SIGGRAPH 2006

Our approach can be combined with these hardware methods

Why is this hard?

Simple analogy:

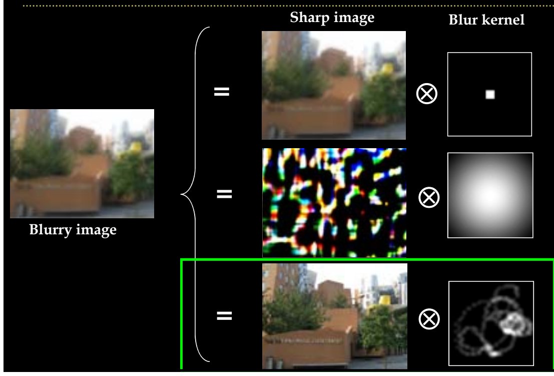
11 is the product of two numbers.
What are they?

No unique solution:

11 = 1 x 11
11 = 2 x 5.5
11 = 3 x 3.667
etc…..

Need more information !!!!

Multiple possible solutions



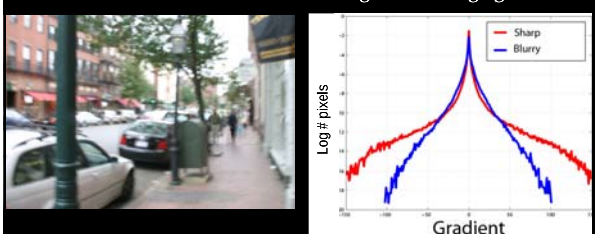
Natural image statistics

Characteristic distribution with heavy tails

Histogram of image gradients

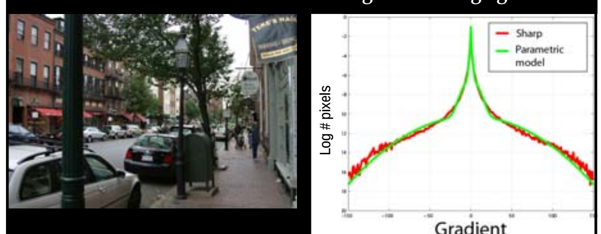
Blurry images have different statistics

Histogram of image gradients



Parametric distribution

Histogram of image gradients



Use parametric model of sharp image statistics

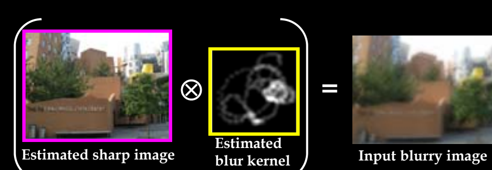
Uses of natural image statistics

- Denoising [Portilla et al. 2003, Roth and Black, CVPR 2005]
- Superresolution [Tappen et al., ICCV 2003]
- Intrinsic images [Weiss, ICCV 2001]
- Inpainting [Levin et al., ICCV 2003]
- Reflections [Levin and Weiss, ECCV 2004]
- Video matting [Apostoloff & Fitzgibbon, CVPR 2005]

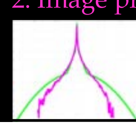
Corruption process assumed known

Three sources of information

1. Reconstruction constraint:



2. Image prior:



Distribution of gradients

3. Blur prior:

Positive & Sparse

Three sources of information

y = observed image b = blur kernel x = sharp image

Three sources of information

y = observed image b = blur kernel x = sharp image

$p(b, x|y)$
Posterior

Three sources of information

y = observed image b = blur kernel x = sharp image

$$p(b, x|y) = k \; p(y|b, x) \; p(x) \; p(b)$$

Posterior — 1. Likelihood (Reconstruction constraint) — 2. Image prior — 3. Blur prior

1. Likelihood $p(y|b, x)$

y = observed image b = blur x = sharp image

Reconstruction constraint:

$$p(y|b, x) = \prod_i N(y_i | x_i \otimes b, \sigma^2)$$

$$\propto \prod_i e^{-\frac{(x_i \otimes b - y_i)^2}{2\sigma^2}}$$

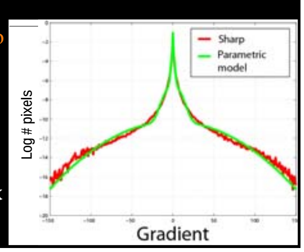
i - pixel index

2. Image prior $p(x)$

y = observed image b = blur x = sharp image

$$p(x) = \prod_i \sum_{c=1}^{C} \pi_c N(f(x_i)|0, s_c^2)$$

Mixture of Gaussians fit to empirical distribution of image gradients



i - pixel index
c - mixture component index
f - derivative filter

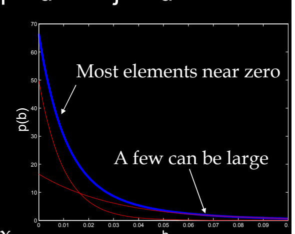
3. Blur prior $p(b)$

y = observed image b = blur x = sharp image

$$p(b) = \prod_j \sum_{d=1}^{D} \pi_d E(b_j | \lambda_d)$$

Mixture of Exponentials

- Positive & sparse
- No connectivity constraint



j - blur kernel element
d - mixture component index

The obvious thing to do

$$p(b|x,y) = k \; p(y|b,x) \; p(x) \; p(b)$$

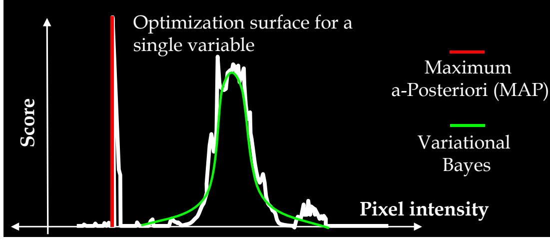
Posterior — 1. Likelihood (Reconstruction constraint) — 2. Image prior — 3. Blur prior

- Combine 3 terms into an objective function
- Run conjugate gradient descent
- This is Maximum a-Posteriori (MAP)

No success!

Variational Bayesian approach

Keeps track of uncertainty in estimates of image and blur by using a distribution instead of a single estimate



Variational Independent Component Analysis

Miskin and Mackay, 2000

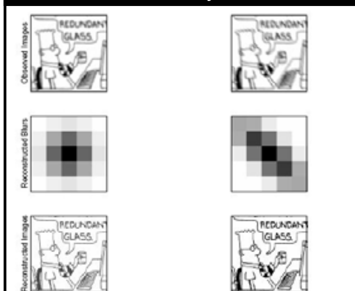


Fig. 9. Demonstration of the deconvolution of two blurred images. In each test the same image was blurred by a different filter. The reconstructed filters match the true filters. The reconstructed images are close to the hidden images. [Dilbert image Copyright©1997 United Feature Syndicate, Inc., used with permission.]

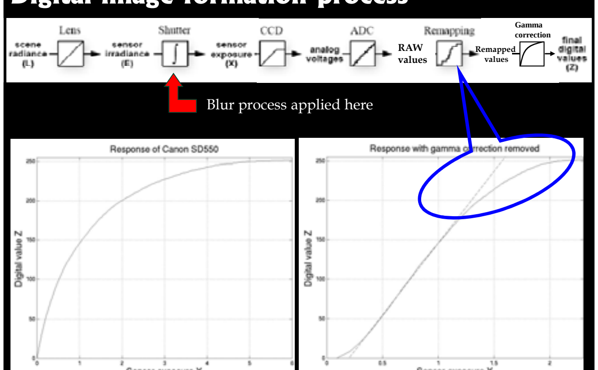
- Binary images
- Priors on intensities
- Small, synthetic blurs
- Not applicable to natural images

Overview of algorithm

1. Pre-processing
2. Kernel estimation
   - Multi-scale approach
3. Image reconstruction
   - Standard non-blind deconvolution routine

Input image

Digital image formation process

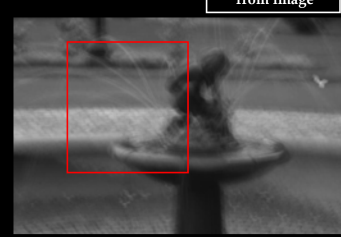


P. Debevec & J. Malik, Recovering High Dynamic Range Radiance Maps from Photographs", SIGGRAPH 97

Preprocessing

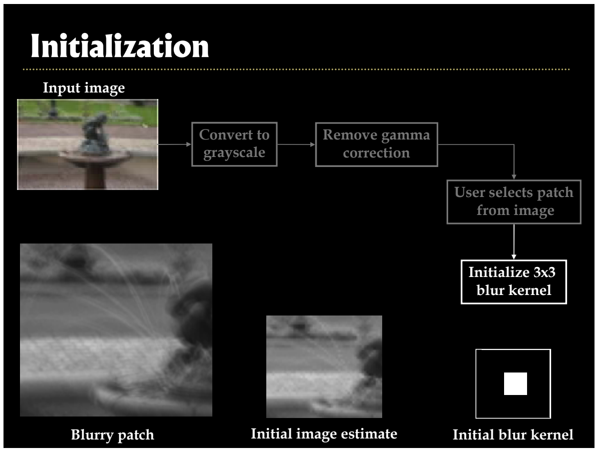
Input image

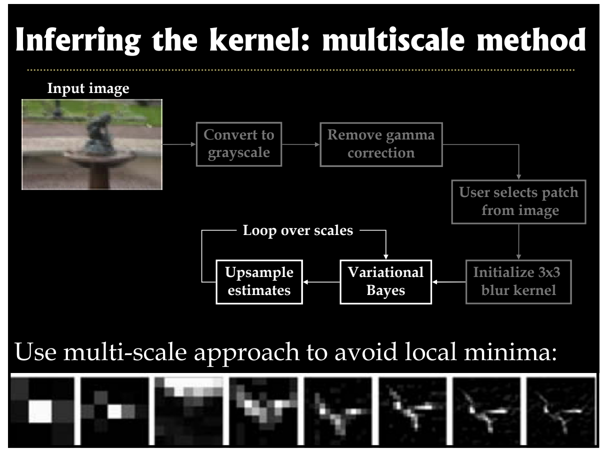
Convert to grayscale → Remove gamma correction → User selects patch from image



Bayesian inference too slow to run on whole image

Infer kernel from this patch


Initialization
Input image
Convert to grayscale
Remove gamma correction
User selects patch from image
Initialize 3x3 blur kernel
Blurry patch
Initial image estimate
Initial blur kernel


Inferring the kernel: multiscale method
Input image
Convert to grayscale
Remove gamma correction
User selects patch from image
Loop over scales
Upsample estimates
Variational Bayes
Initialize 3x3 blur kernel
Use multi-scale approach to avoid local minima:

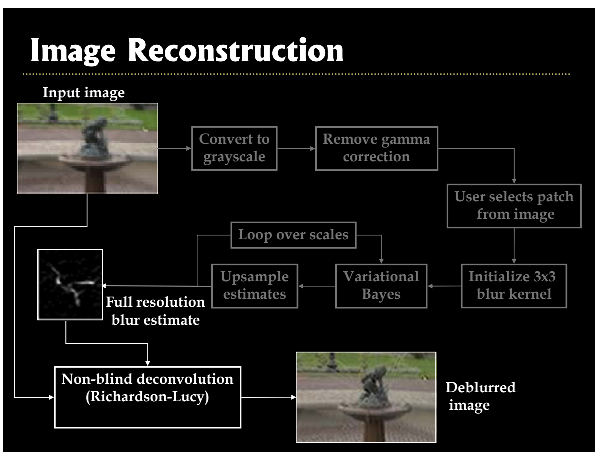
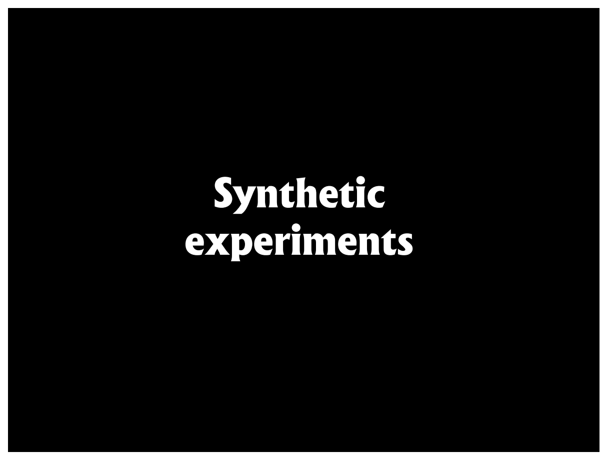
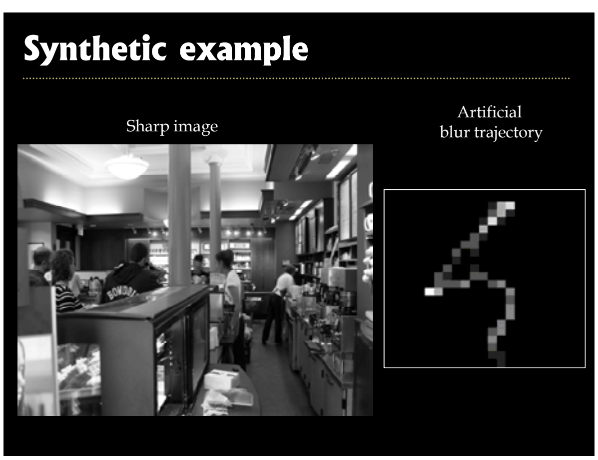
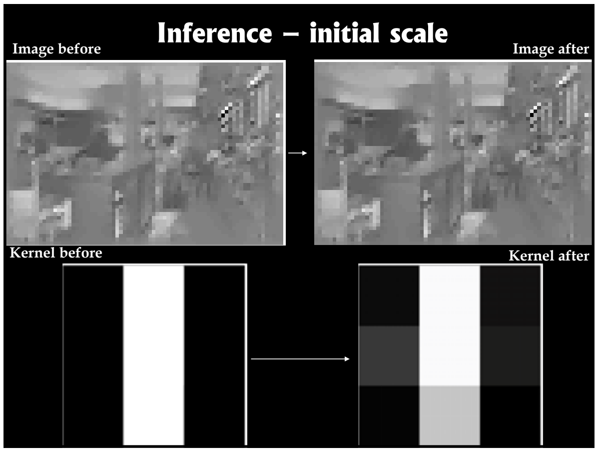


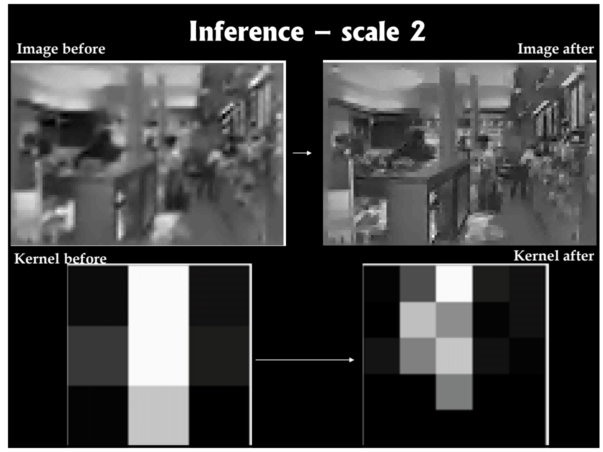
Image Reconstruction
Input image
Convert to grayscale
Remove gamma correction
User selects patch from image
Loop over scales
Upsample estimates
Variational Bayes
Initialize 3x3 blur kernel
Full resolution blur estimate
Non-blind deconvolution (Richardson-Lucy)
Deblurred image

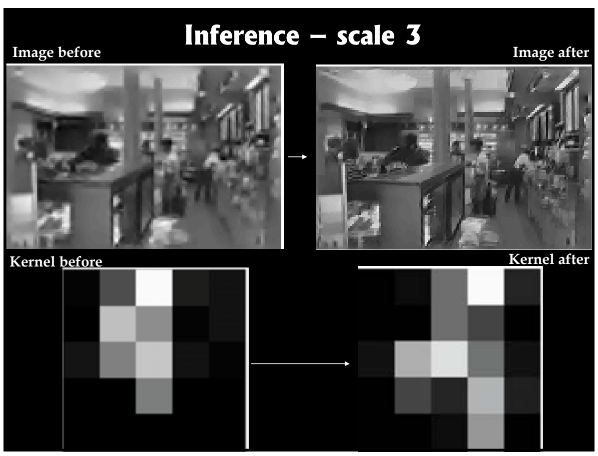

Synthetic experiments

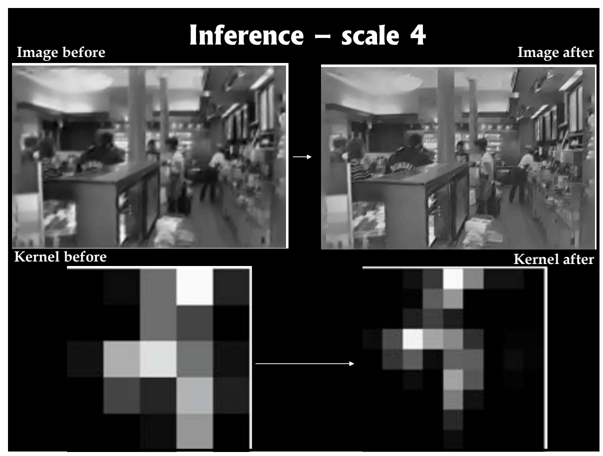

Synthetic example
Sharp image
Artificial blur trajectory

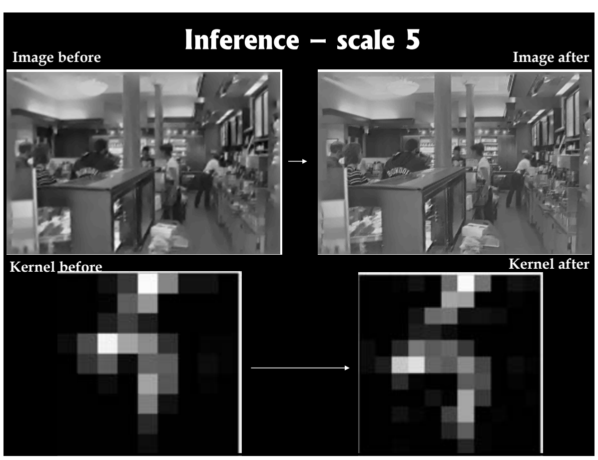

Synthetic blurry image

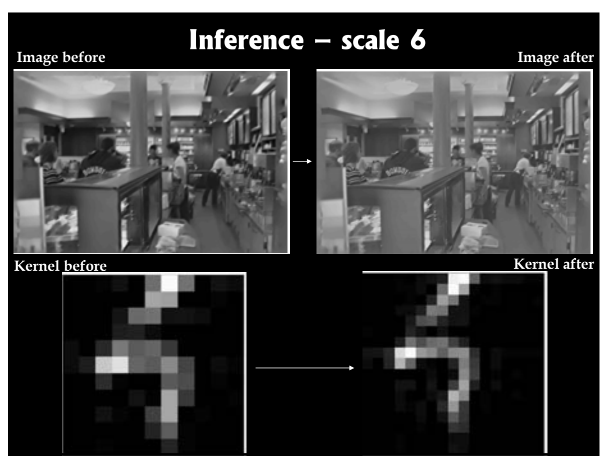

Inference – initial scale
Image before
Image after
Kernel before
Kernel after

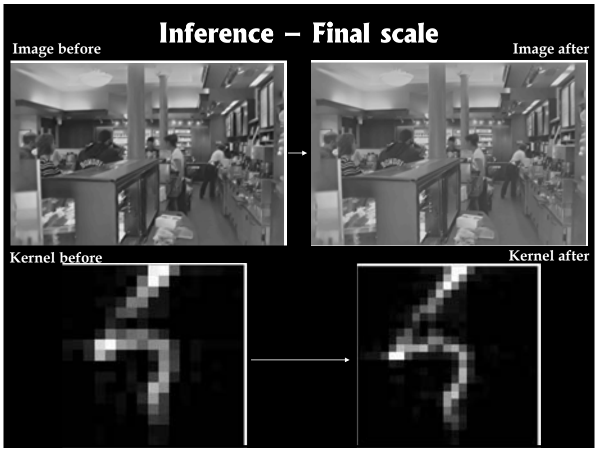

Inference – scale 2
Image before
Image after
Kernel before
Kernel after

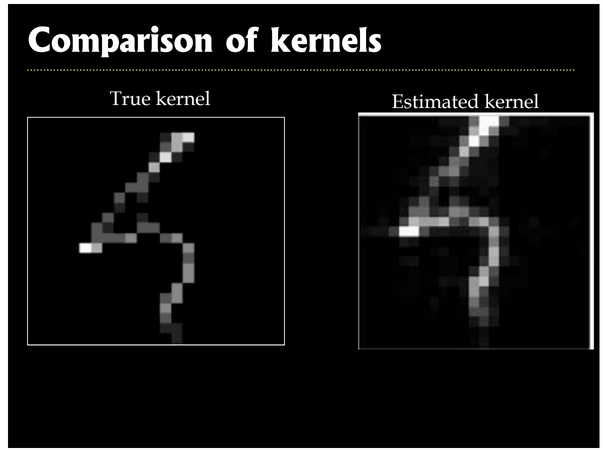

Inference – scale 3
Image before
Image after
Kernel before
Kernel after


Inference – scale 4
Image before
Image after
Kernel before
Kernel after


Inference – scale 5
Image before
Image after
Kernel before
Kernel after


Inference – scale 6
Image before
Image after
Kernel before
Kernel after

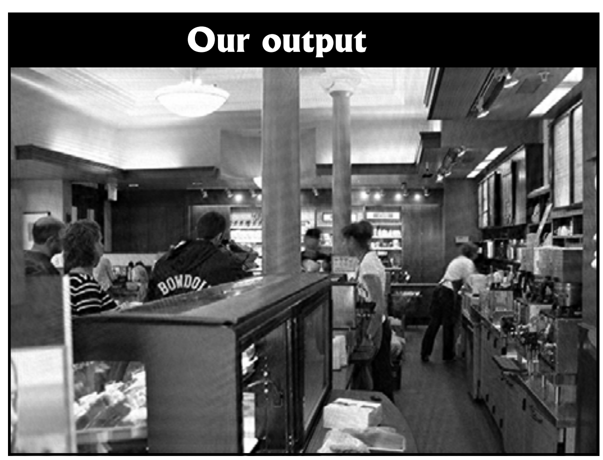

Inference – Final scale
Image before
Image after
Kernel before
Kernel after


Comparison of kernels
True kernel
Estimated kernel


Blurry image


Matlab's deconvblind


Blurry image


Our output

True sharp image

What we do and don't model

DO
- Gamma correction
- Tone response curve (if known)

DON'T
- Saturation
- Jpeg artifacts
- Scene motion
- Color channel correlations

Real experiments

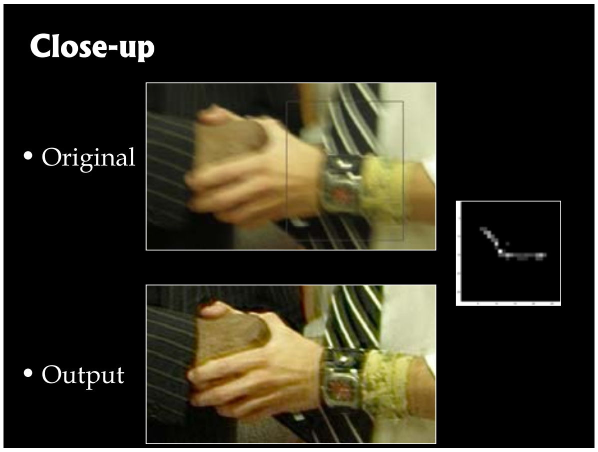
Results on real images

Submitted by people from their own photo collections

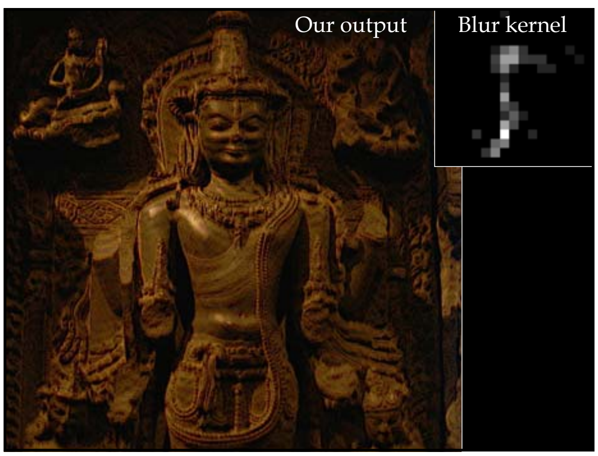
Type of camera unknown

Output does contain artifacts
- Increased noise
- Ringing

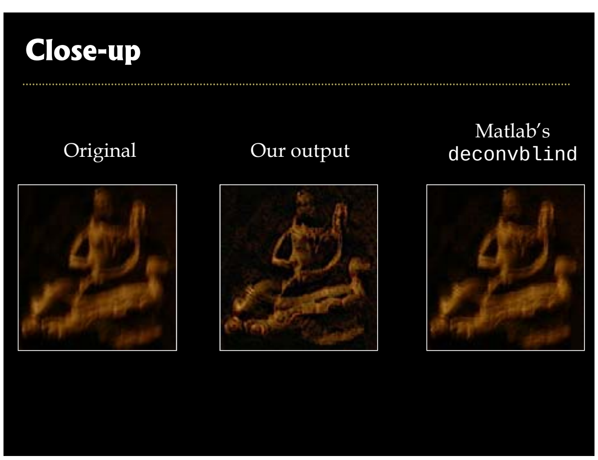
Compare with existing methods


Close-up
• Original
• Output

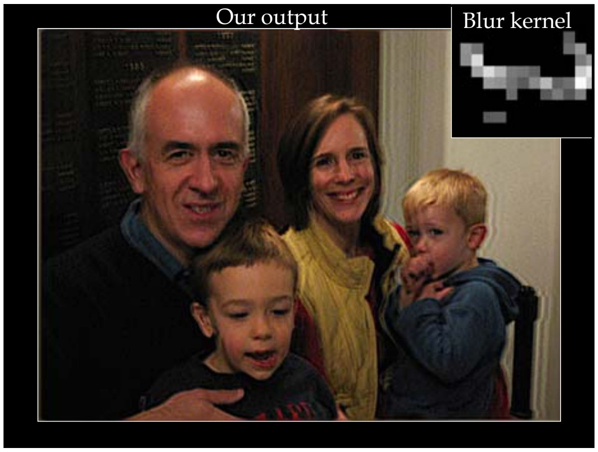

Original photograph


Our output
Blur kernel

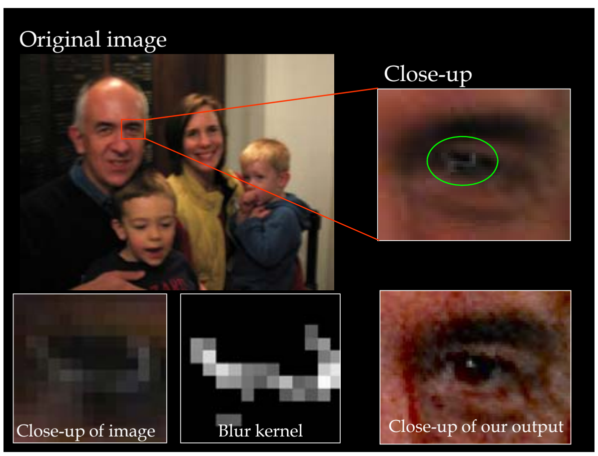

Matlab's deconvblind


Close-up
Original
Our output
Matlab's
deconvblind


Original photograph


Our output
Blur kernel


Photoshop sharpen more

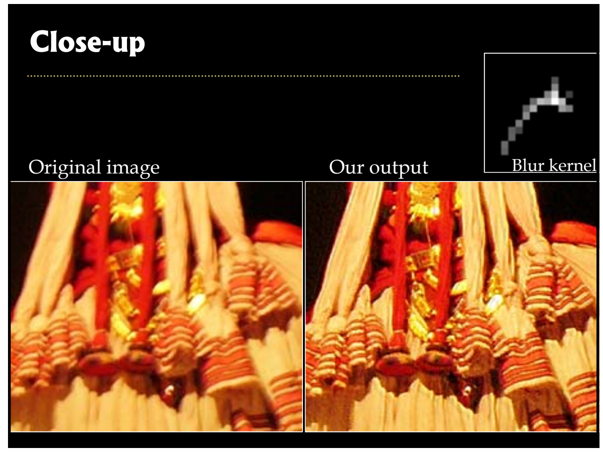

Original image
Close-up
Close-up of image
Blur kernel
Close-up of our output


Original photograph
睇到女人定係
愛斯基摩人?

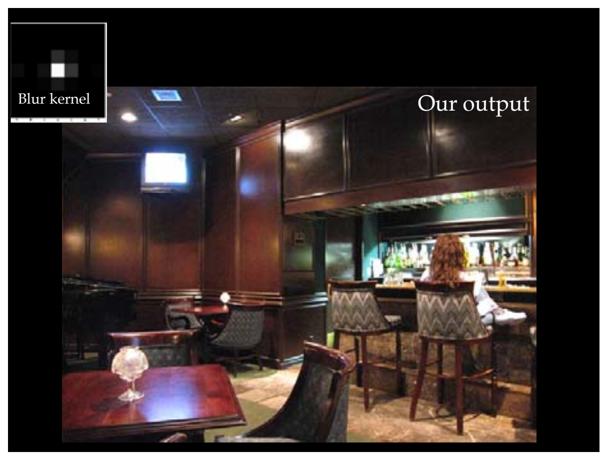

Our output
Blur kernel
睇到女人定係
愛斯基摩人?


Original image


Our output
Blur kernel

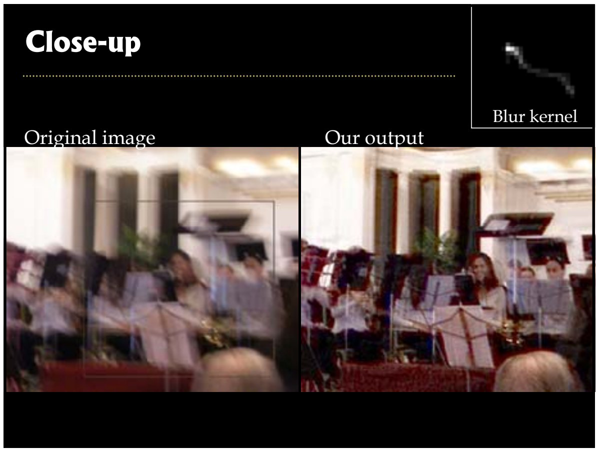

Close-up
Original image
Our output
Blur kernel


What about a sharp image?
Original photograph

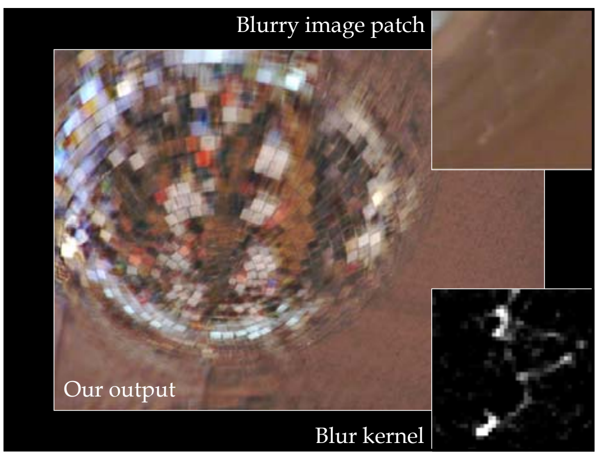

Blur kernel
Our output


Original photograph

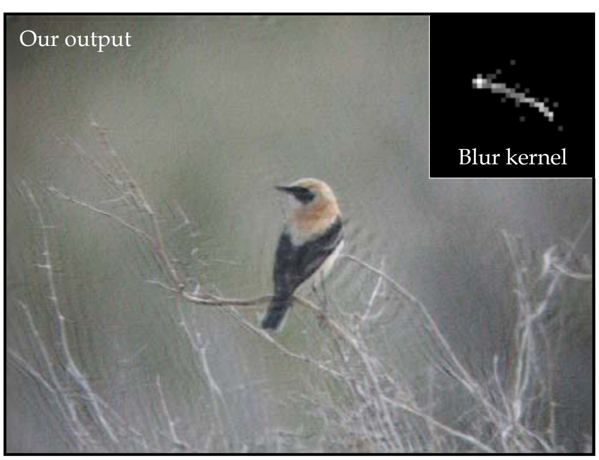

Blur kernel
Our output

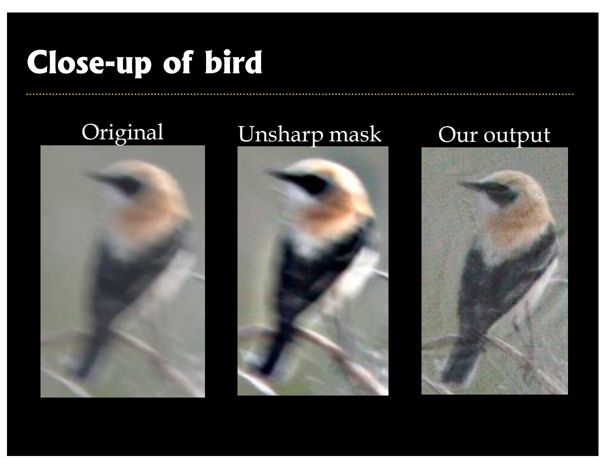

Close-up
Blur kernel
Original image
Our output


Original photograph


Blurry image patch
Our output
Blur kernel


Original photograph


Our output
Blur kernel


Close-up of bird
Original
Unsharp mask
Our output

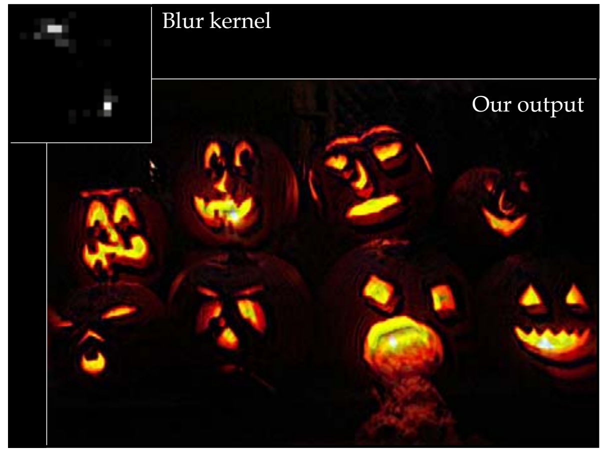


Image artifacts & estimated kernels

Blur kernels

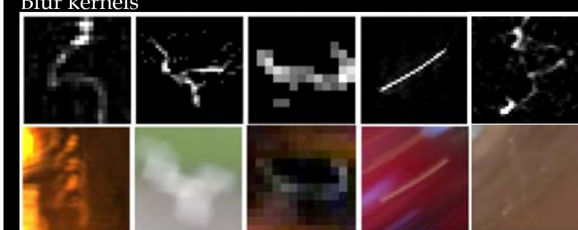


Image patterns

Note: blur kernels were inferred from large image patches, NOT the image patterns shown

Code available online

http://cs.nyu.edu/~fergus/research/deblur.html

Summary

Method for removing camera shake from real photographs

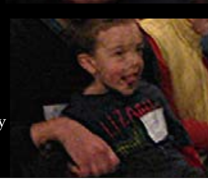
First method that can handle complicated blur kernels

Uses natural image statistics

Non-blind deconvolution currently simplistic

Things we have yet to model:

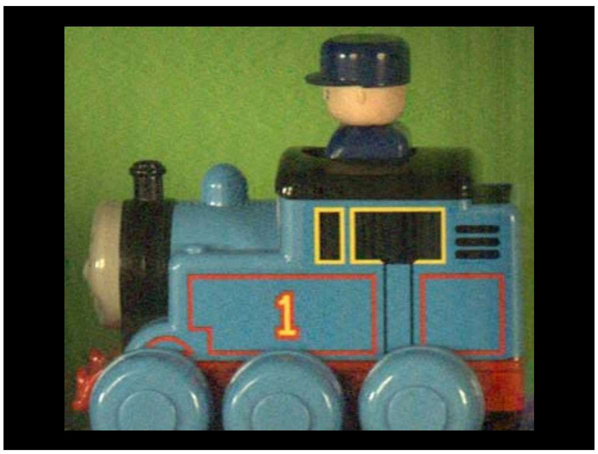
- Correlations in colors, scales, kernel continuity
- JPEG noise, saturation, object motion

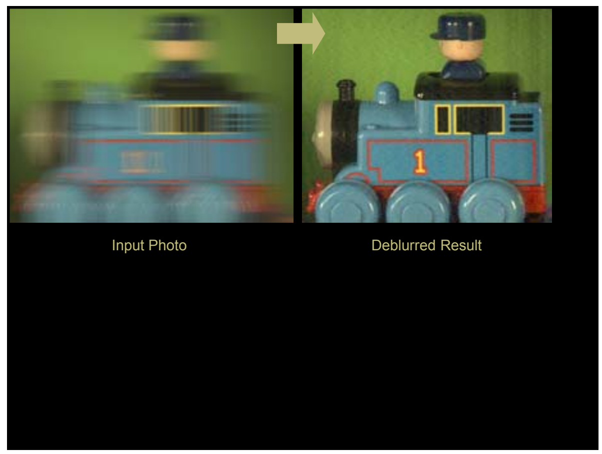


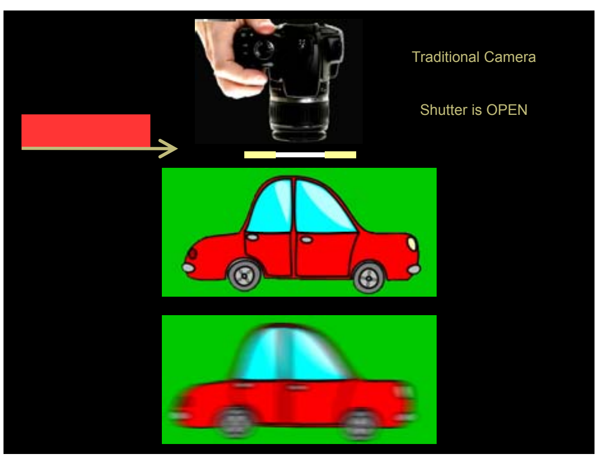
Overview

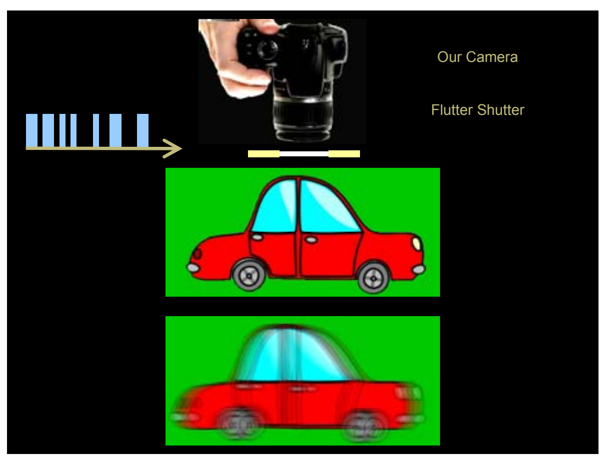
- Removing Camera Shake
  - Non-blind
  - Blind
- Removing Motion Blur
  - Non-blind
  - Blind

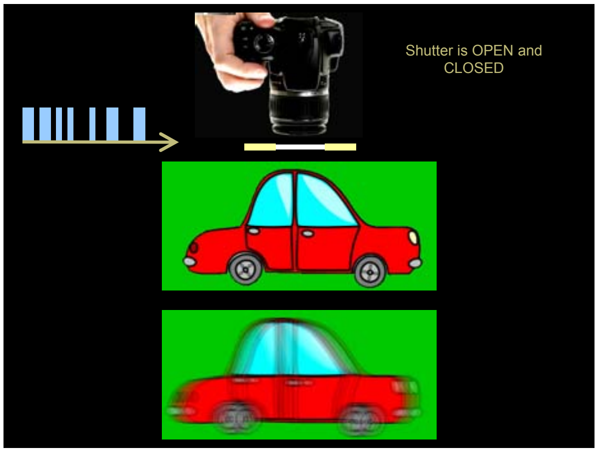

Fluttered Shutter Camera
Raskar, Agrawal, Tumblin Siggraph2006
Ferroelectric shutter in front of the lens is turned
opaque or transparent in a rapid binary sequence

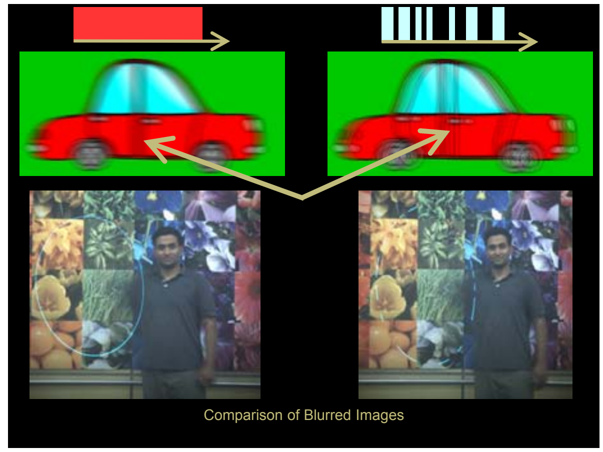


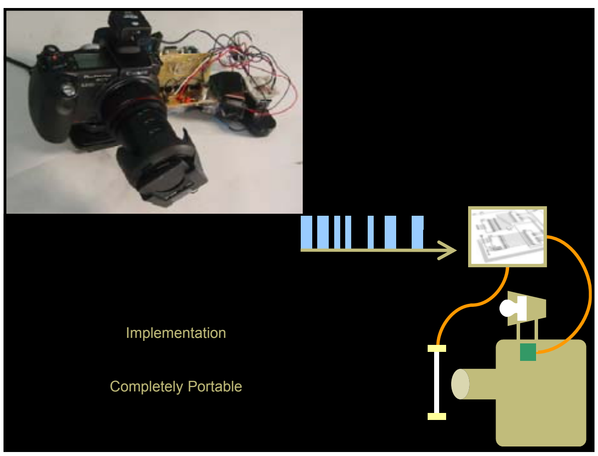

Input Photo
Deblurred Result

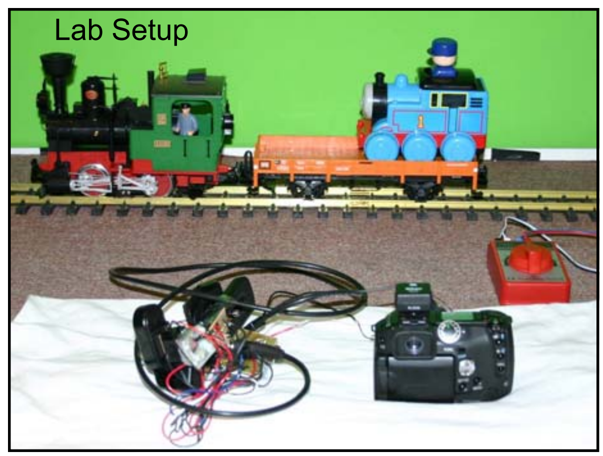

Traditional Camera
Shutter is OPEN

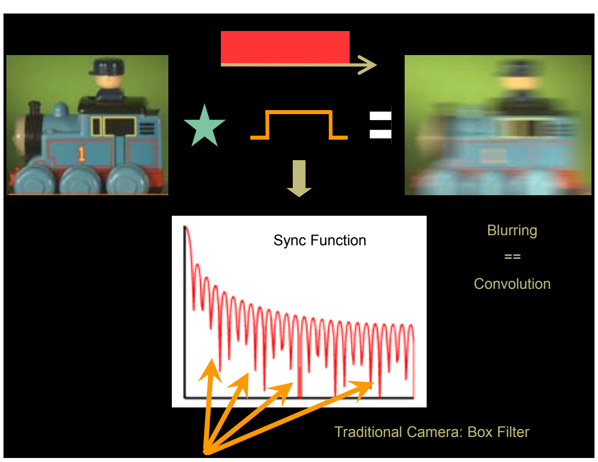

Our Camera
Flutter Shutter

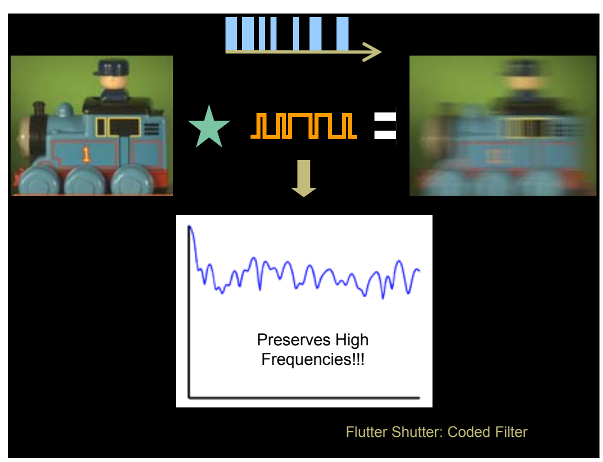

Shutter is OPEN and
CLOSED

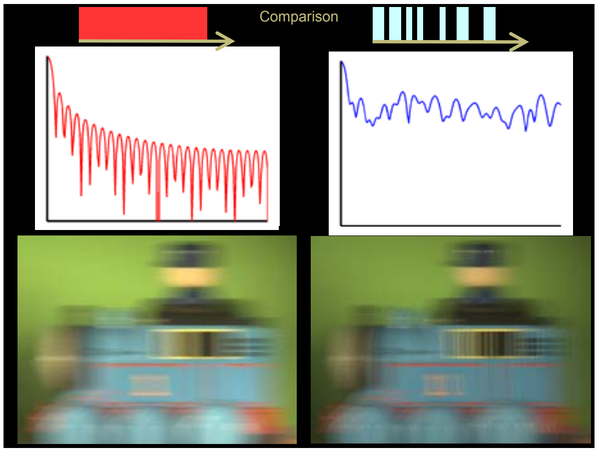

Comparison of Blurred Images

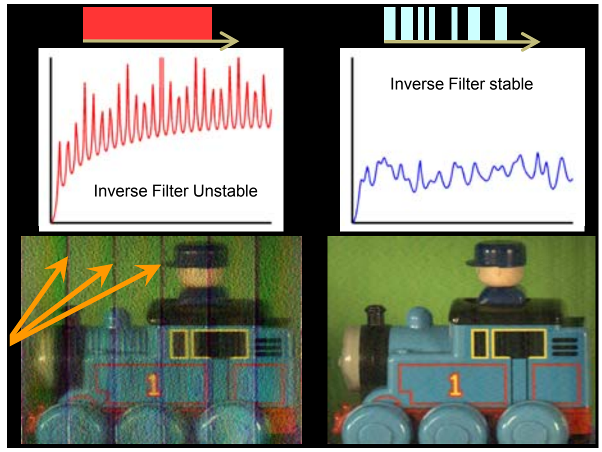

Implementation
Completely Portable

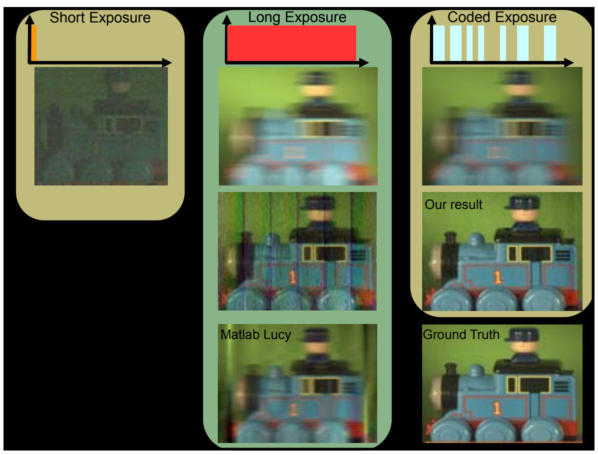

Lab Setup


Sync Function
Blurring
==
Convolution
Traditional Camera: Box Filter


Preserves High
Frequencies!!!
Flutter Shutter: Coded Filter







Overview

- Removing Camera Shake
  - Non-blind
  - Blind
- Removing Motion Blur
  - Non-blind
  - Blind

Blind motion deblurring using image statistics

Anat Levin
School of Computer Science and Engineering
The Hebrew University of Jerusalem

Use statistics to determine blur size

- Assumes direction of blur known

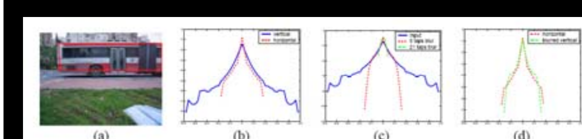


Figure 1: Blurred versus unblurred derivatives histograms. (a) Input image. (b) Horizontal derivatives within the blurred region versus vertical derivatives in the entire image. (c) Simulating different blurs in the vertical direction. (d) Horizontal derivatives within the blurred region matched with blurred verticals (4 tap blur).

Input image

Deblur whole image at once

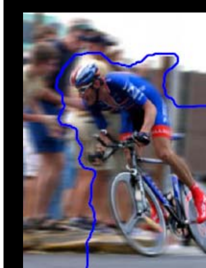
Local Evidence

Vertical edges map and the maximum likelihood model in each pixel

White: $lk_{\text{unblurred}}(i) > lk_{\text{12 pixels blur}}(i)$

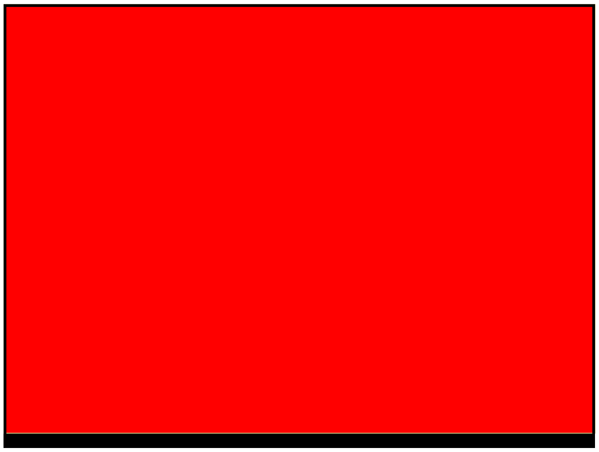
Gray: $lk_{\text{unblurred}}(i) < lk_{\text{12 pixels blur}}(i)$

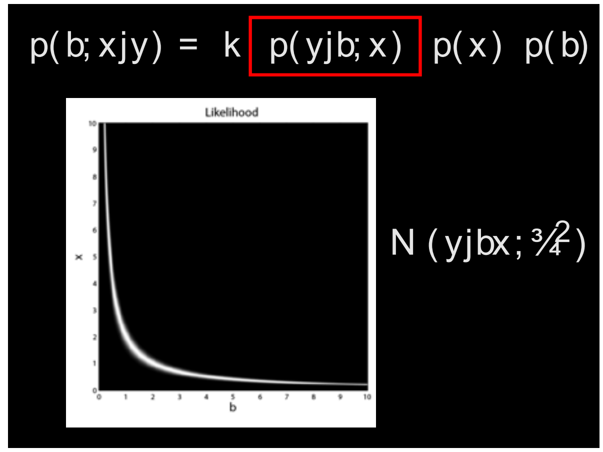
Proposed boundary

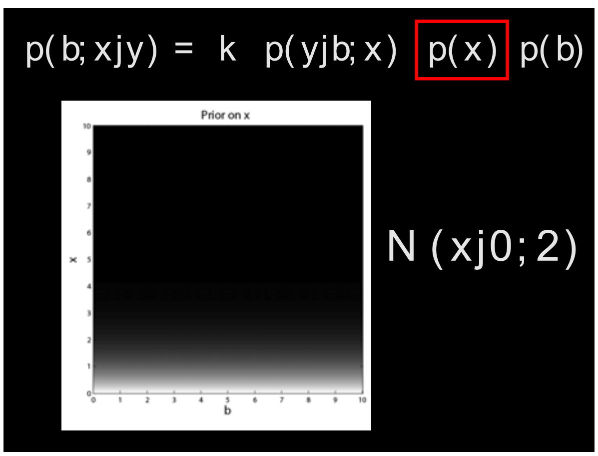


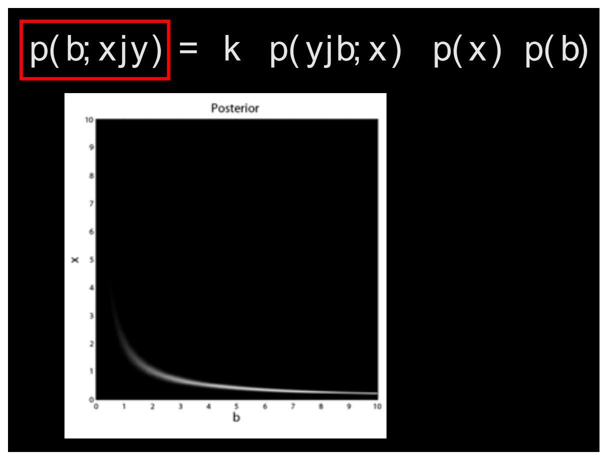
Result image

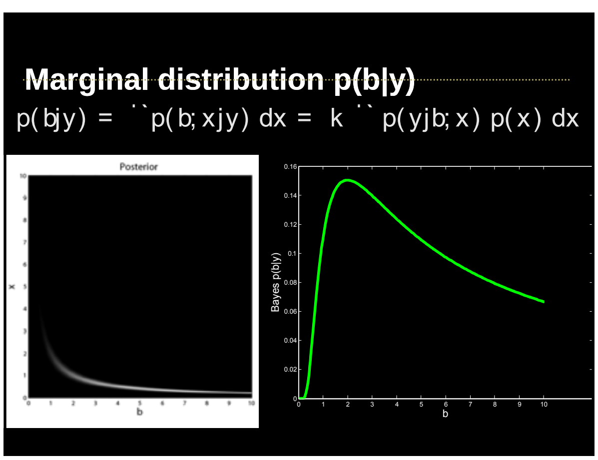
Input image (for comparison)

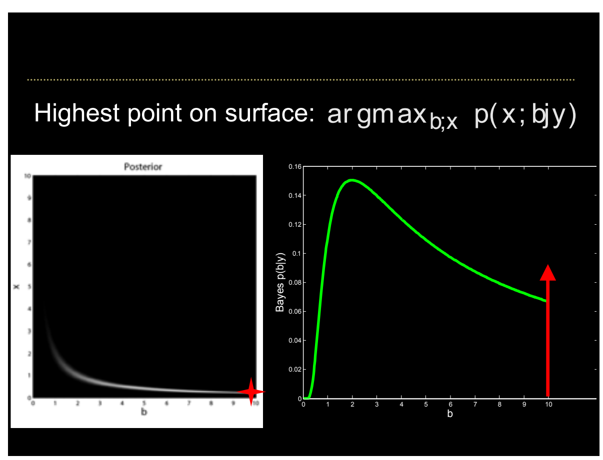



p(b; xjy) = k p(yjb; x) p(x) p(b)
Likelihood
x
b
N (yjbx; ¾2)

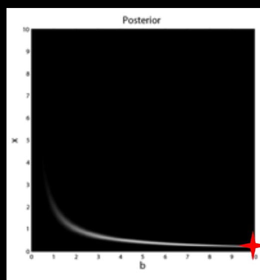

p(b; xjy) = k p(yjb; x) p(x) p(b)
Prior on x
x
b
N (xj0; 2)


p(b; xjy) = k p(yjb; x) p(x) p(b)
Posterior
x
b


Marginal distribution p(b|y)
p(bjy) = ``p( b; xjy) dx = k `` p(yjb; x) p(x) dx
Posterior
x
b
Bayes p(b|y)

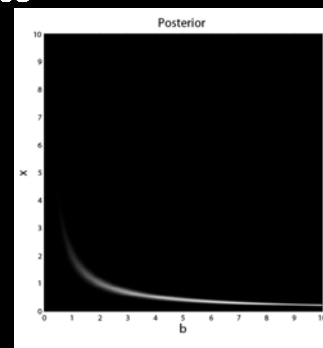

Highest point on surface: ar gmax b;x p(x; bjy)
Posterior
x
b
Bayes p(b|y)

Highest point on surface: $\arg\max_{b;x}\ p(x;b|y)$



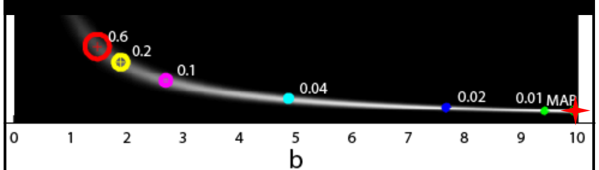
Variational Bayes

- True Bayesian approach not tractable
- Approximate posterior with simple distribution

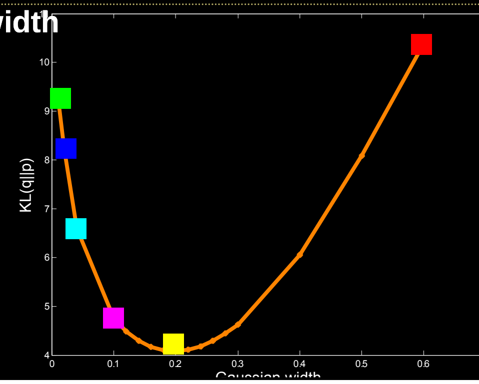


Fitting posterior with a Gaussian

- Approximating distribution $q(x;b)$ is Gaussian
- Minimize $KL(q(x;b)\,||\,p(x;b|y))$

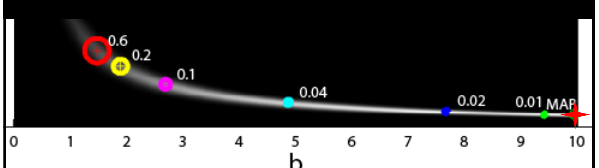


KL-Distance vs Gaussian width

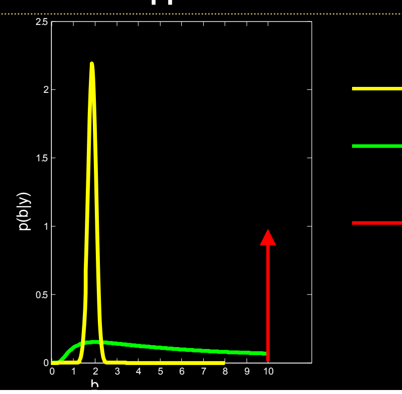


Fitting posterior with a Gaussian

- Approximating distribution $q(x;b)$ is Gaussian
- Minimize $KL(q(x;b)\,||\,p(x;b|y))$

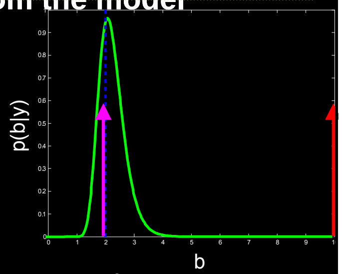


Variational Approximation of Marginal



Try sampling from the model

Let true b = 2



Repeat:

- Sample x ~ N(0,2)
- Sample n ~ N(0,σ^2)
- y = xb + n
- Compute $p_{MAP}(b|y)$, $p_{Bayes}(b|y)$ & $p_{Variational}(b|y)$
- Multiply with existing density estimates (assume iid)

Setup of Variational Approach

x - b= y! r x - b= r y

Approximate posterior p(r x; bjr y)
with q(r x; b)

q(r x; b) = q(r x) q(b)

q(r x)

q(b)

K L (q(r x) q(b) jj p(r x; bjr y))