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
Let’s take a photo

Blurry result
Slow-motion replay
Slow-motion replay

Motion of camera
Image formation process

Blurred image = Convolution operator = Sharp image

- **Blurry image**: Input to algorithm
- **Sharp image**: Desired output
- **Model is approximation**
- **Assume static scene**

- **Blur kernel**: Convolution operator
Existing work on image deblurring

Old problem:

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

![](image1.png)

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

\[
\text{Blurry image} = \star \text{Sharp image} = \star \text{Blur kernel}
\]
Natural image statistics

Characteristic distribution with heavy tails

Histogram of image gradients
Blurry images have different statistics.

Histogram of image gradients.
Parametric distribution

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:

3. Blur prior:

Estimated sharp image \times Estimated blur kernel = Input blurry image

Distribution of gradients

Positive & Sparse
Three sources of information

\( y = \text{observed image} \quad b = \text{blur kernel} \quad x = \text{sharp image} \)
Three sources of information

\[ y = \text{observed image} \quad b = \text{blur kernel} \quad x = \text{sharp image} \]

\[ p(b, x|y) \]

\textit{Posterior}
Three sources of information

\[ p(b, x | y) = k \]

1. Likelihood (Reconstruction constraint)

2. Image prior

3. Blur prior

\[ p(y | b, x) \]

\[ p(x) \]

\[ p(b) \]

\( y = \) observed image \hspace{1cm} \( b = \) blur kernel \hspace{1cm} \( x = \) sharp image
1. Likelihood \( p(y| b, x) \)

\[
\begin{align*}
    y &= \text{observed image} \\
    b &= \text{blur} \\
    x &= \text{sharp image}
\end{align*}
\]

Reconstruction constraint:

\[
p(y| b, x) = \prod_i \mathcal{N}(y_i | x_i \otimes b, \sigma^2)
\]

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

\( i \) - pixel index
2. Image prior \( p(x) \)

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

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 $\quad b =$ blur $\quad x =$ sharp image

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

Mixture of Exponentials
- Positive & sparse
- No connectivity constraint

$j -$ blur kernel element $\quad d -$ mixture component index
The obvious thing to do

\[ p(b, x | y) = k \quad p(y | b, x) \quad p(x) \quad p(b) \]

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.

Optimization surface for a single variable.

- Maximum a-Posteriori (MAP)
- Variational Bayes

Pixel intensity

Score
Variational Independent Component Analysis
Miskin and Mackay, 2000

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

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.]
Setup of Variational Approach

Work in gradient domain:

\[ x \otimes b = y \rightarrow \nabla x \otimes b = \nabla y \]

Approximate posterior \( p(\nabla x, b | \nabla y) \)
with \( q(\nabla x, b) \)

Assume \( q(\nabla x, b) = q(\nabla x)q(b) \)

\( q(\nabla x) \) is Gaussian on each pixel
\( q(b) \) is rectified Gaussian on each blur kernel element

Cost function \( KL(q(\nabla x)q(b) \| p(\nabla x, b | \nabla y)) \)
Overview of algorithm

1. Pre-processing

2. Kernel estimation
   - Multi-scale approach

3. Image reconstruction
   - Standard non-blind deconvolution routine
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

1. Input image
2. Convert to grayscale
3. Remove gamma correction
4. User selects patch from image
5. Initialize 3x3 blur kernel
6. Loop over scales
   - Upsample estimates
   - Variational Bayes

Use multi-scale approach to avoid local minima:
Image Reconstruction

1. **Input image**
2. **Convert to grayscale**
3. **Remove gamma correction**
4. **User selects patch from image**
5. **Loop over scales**
6. **Upsample estimates**
7. **Variational Bayes**
8. **Initialize 3x3 blur kernel**
9. **Non-blind deconvolution (Richardson-Lucy)**
10. **Full resolution blur estimate**
11. **Deblurred image**
Synthetic experiments
Is blur kernel really stationary?

8 different people, handholding camera, using 1 second exposure
Dots from each corner

Person 1

Top left

Top right

Bot. left

Bot. right

Person 2

Person 3

Person 4
Synthetic example

Sharp image

Artificial blur trajectory
Synthetic blurry image
Inference – initial scale

Image before

Kernel before

Kernel after

Image after
Inference – scale 2

Image before

Kernel before

Image after

Kernel after
Inference – scale 3
Inference – scale 4

Image before

Kernel before

Image after

Kernel after
Inference – scale 5

Image before

Kernel before

Image after

Kernel after
Comparison of kernels

True kernel

Estimated kernel
Our output
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
Matlab’s deconvblind
Close-up

Original

Our output

Matlab’s deconvblind
Original photograph
Photoshop sharpen more
Original photograph
Our output
Photoshop “Smart Sharpen”

Blur kernel
Matlab's `deconvblind`
睇到女人定係愛斯基摩人？
Our output

睇到女人定係
愛斯基摩人？

全新地鐵「車箱橫額廣告」
讓您睇得更多！

Blur kernel
Our output

Blur kernel
Close-up of AV equipment

Original photograph

Our output
What about a sharp image?

Original photograph
Close-up

• Original

• Output
Our output
Close-up

Original image

Our output

Blur kernel
Original photograph
Our output

Blur kernel
Close-up of bird

Original

Unsharp mask

Our output
Blur kernel

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://people.csail.mit.edu/fergus/research/deblur.html

Matlab source code

Please fill in the form here and I will email you the code.
Here is the accompanying README file.
Summary

First method that can handle complicated real-world blur kernels

Results still contain artifacts

Big leap on an old, hard problem

Many things to improve:

- Non-blind deconvolution, saturation etc.
Digital image formation process

- Lens: scene radiance (L) → sensor irradiance (E)
- Shutter: sensor exposure (X)
- CCD: analog voltages
- ADC: RAW values → Remapped values
- Remapping: Remapped values → Gamma correction
- Final digital values (Z)

Blur process applied here

Response of Canon SD550

P. Debevec & J. Malik, Recovering High Dynamic Range Radiance Maps from Photographs®, SIGGRAPH 97
Simple 1-D example

\[ y = bx + n \]

y = observed image
b = blur
x = sharp image
n = noise \( \sim N(0, \sigma^2) \)

Let \( y = 2 \)
\[ p(b, x|y) = k p(y|b, x) p(x) p(b) \]

Let \( y = 2 \)
\[ \sigma^2 = 0.1 \]
\[ \mathcal{N}(y|bx, \sigma^2) \]
\[ p(b, x \mid y) = k \quad p(y \mid b, x) \] 
\[ p(x) \] 
\[ p(b) \]

Gaussian distribution:

\[ \mathcal{N}(x \mid 0, 2) \]
\[ p(b, x | y) = k \ p(y | b, x) \ p(x) \ p(b) \]
Marginal distribution $p(b|y)$

$$p(b|y) = \int p(b, x|y) \, dx = k \int p(y|b, x) \, p(x) \, dx$$
MAP solution

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 $KL(q(x, b) \parallel p(x, b|y))$
KL-Distance vs Gaussian width

![Graph showing KL-Distance vs Gaussian width](image-url)
Variational Approximation of Marginal

$\text{Variational}$

$\text{True}$

$marginal$

$\text{MAP}$

$p(b|y)$

$0$ $1$ $2$ $3$ $4$ $5$ $6$ $7$ $8$ $9$ $10$
Try sampling from the model

Let true \( b = 2 \)

Repeat:
- Sample \( x \sim N(0,2) \)
- Sample \( n \sim N(0,\sigma^2) \)
- \( y = xb + n \)
- Compute \( p_{MAP}(b|y) \), \( p_{Bayes}(b|y) \) & \( p_{Variational}(b|y) \)
- Multiply with existing density estimates (assume iid)
Close-up of child

Original photograph

Our output