# A Complete Survey of Novel Deconvolution Algorithm

PingYi.Song

# Outline

- Convolution Model
- Challenge
- Non-Blind Deconvolution
  - Frequency domain
  - Spatial domain
  - Optimization method
- Blind Deconvolution

# Prior knowledge

- Basic image Processing
- Frequency domain
  - Fourier Transform
  - Complex Number
- Basic Linear Algebra

# Convolution Model

$$B = K \otimes I + N$$

- K: point spread function (PSF)
- I: original image
- N: noise
- B : blurry image
- $\otimes$: convolution operator

# Point Spread Function

- Shift invariant
- Shift variant

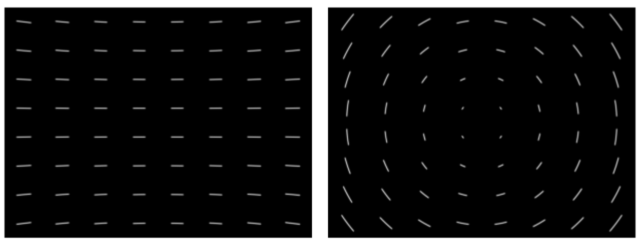


Figure 2. **The paths followed by image points under single-axis rotations.** Left: Rotation about the $Y$-axis. Right: Rotation about the $Z$-axis. Under single-axis camera rotations, the paths followed by points in the image are visibly curved and non-uniform across the image. The focal length of the camera in this simulation is equal to the width of the image, the principal point is at the image's center, and the pixels are assumed to be square.

*Non-uniform Deblurring for Shaken images CVPR 2010, Oliver et al.*


**Figure 6:** *Visualization of ground-truth spatially-varying PSFs: For the blurry image on top, we show a sparsely sampled visualization of the blur kernels across the image plane. There is quite a significant variation across the image plane. To demonstrate the importance of accounting for spatially variance, in the bottom row we show a result where we have deconvolved using the PSF for the correct part of the image and a non-corresponding area.*

*image Deblurring using inertial Measurement Sensors SiGGRAPH 2010, Joshi et al.*

# Noise

- Noise distribution model
  - Gaussian
  - Poisson

$$N = B - K \otimes I$$

# Deconvolution

- Inverse Convolution
- Noise is unknown
  - Unable to recovery true input image
  - Lose texture detail
  - Cause undesirable artifact

$$B = K \otimes I$$

$$I = G \otimes B$$

# Deconvolution

- Deconvolution type:
  - Non-Blind deconvolution: PSF is know

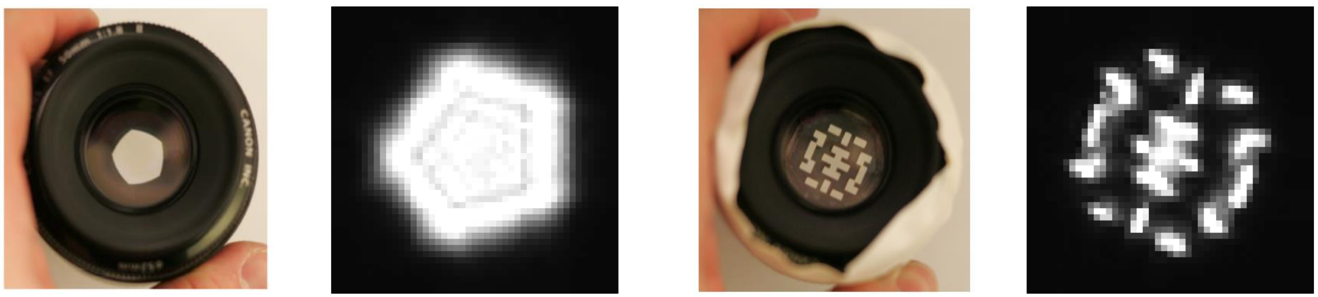
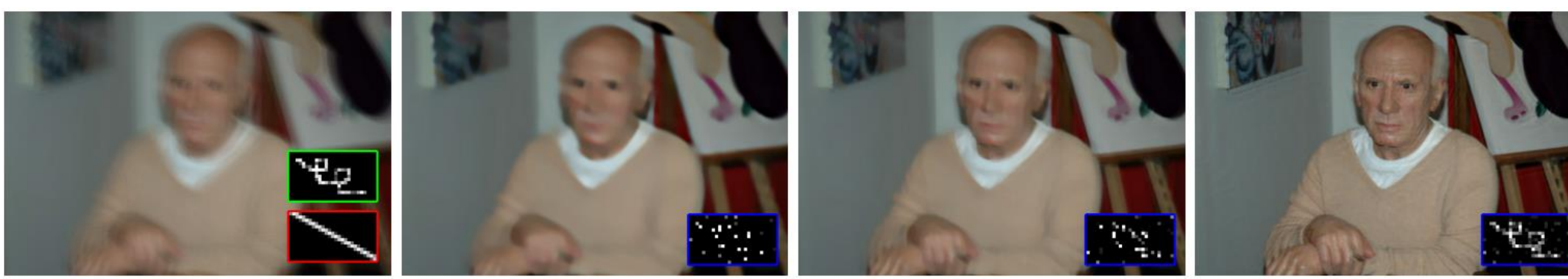


*image and Depth from a Conventional Camera with a Coded Aperture SiGGRAPH07, Levin et al.*

  - Blind deconvolution: PSF is unknown



(a) Blurred image (b) Iteration 1 (c) Iteration 6 (d) Iteration 10

*High-quality Motion Deblurring from a Single image SiGGRAPH 2008, Shan et al.*

# CHALLENGE

# ill-posed problem

- Blind deconvolution
  - 11 = 1 x 11
  - 11 = 2 x 5.5
  - 11 = 3 x 3.667
  - Etc…

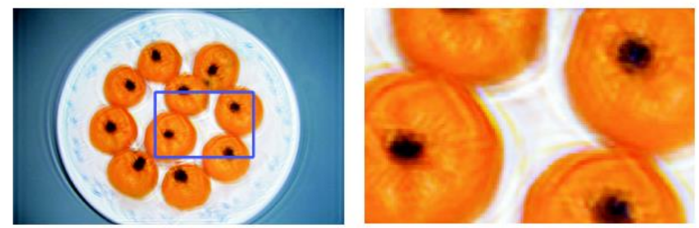
$$B = K \otimes I$$

- Need good constrain/priors

*Old and New Algorithms for Blind Deconvolution*
*Yair Weiss.*

# Ringing Artifacts

- Ringing from strong edges
  – Gibbs phenomenon
- Ringing due to
  – Noise
  – PSF error
  – Lost of image boundary information (easy to eliminate)



(a) (b)

**Figure 2** Ringing artifacts in image deconvolution. (a) A blind deconvolution result. (b) A magnified patch from (a). Ringing artifacts are visible around strong edges.

*High-quality Motion Deblurring from a Single image SiGGRAPH 2008, Shan et al.*

# Point Spread Function Priors

- Priors
  - Sparse (most values close to zero)
  - Positive
  - Sum to 1

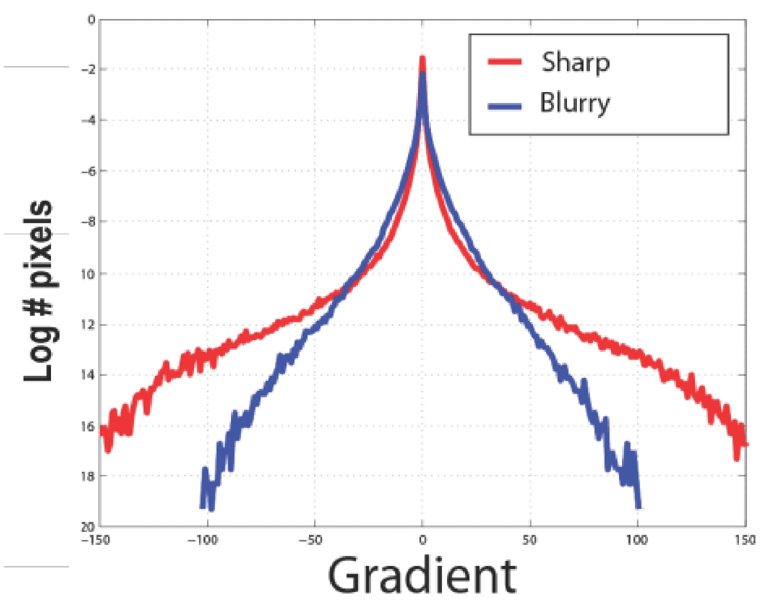
$$\sum_i K_i = 1, K_i \geq 0$$

- Constrains

$$\min \|K\|^{\alpha} \begin{cases} \min \sum_i |K_i|, \alpha = 1 \\ \min \sum_i \lambda e^{-\lambda K_i}, \alpha < 1 \end{cases}$$

# Image Priors

- Natural image statistics
  - Histogram of image gradients
- Sparse derivatives in different orders
  - Prewitt (1st-order)
  - Laplacian (2nd-order)
- Total variation



# Blur Type

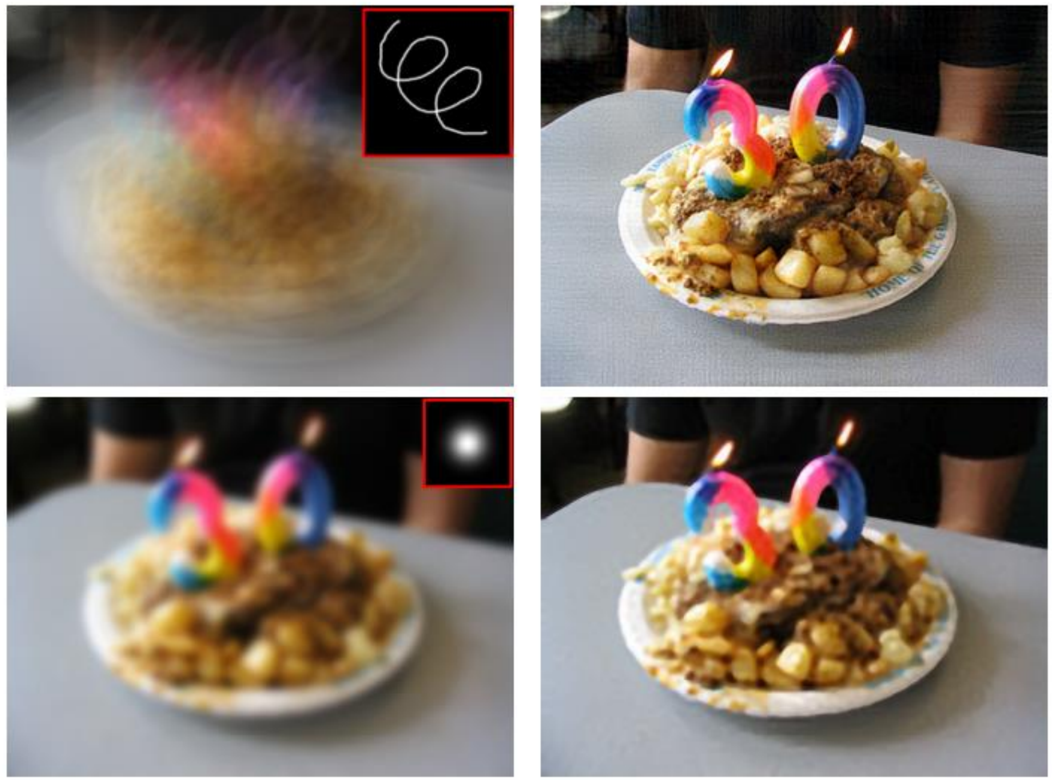


**Figure 15:** *Limitations. Top: our result on a* $160 \times 160$ *motion blur kernel. Boundary artifacts appear. Bottom: our result on a* $40 \times 40$ *gaussian kernel. High frequency details destroyed in the blurring process cannot be recovered.*

*Progressive Inter-scale and Intra-scale Non-blind Image Deconvolution SIGGRAPH 2008.*

Naïve method and Wiener Filter

# DECONVOLUTION IN FREQUENCY DOMAIN

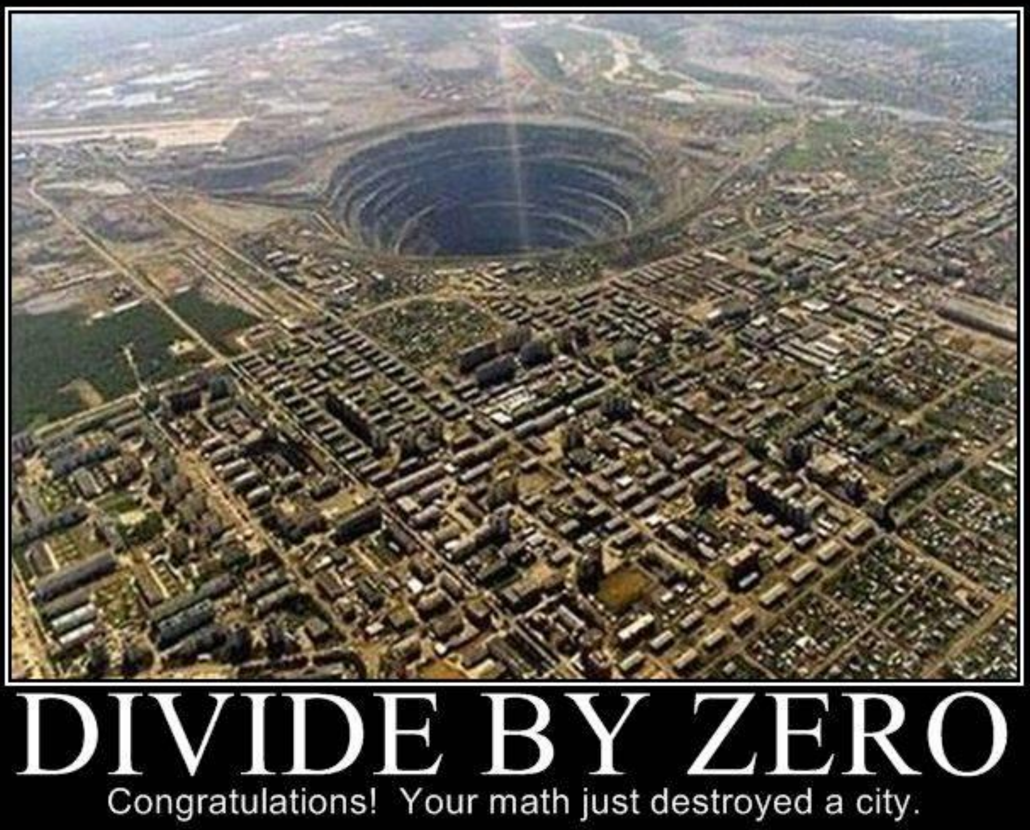
# Naïve Deconvolution

$$B = K \otimes I$$

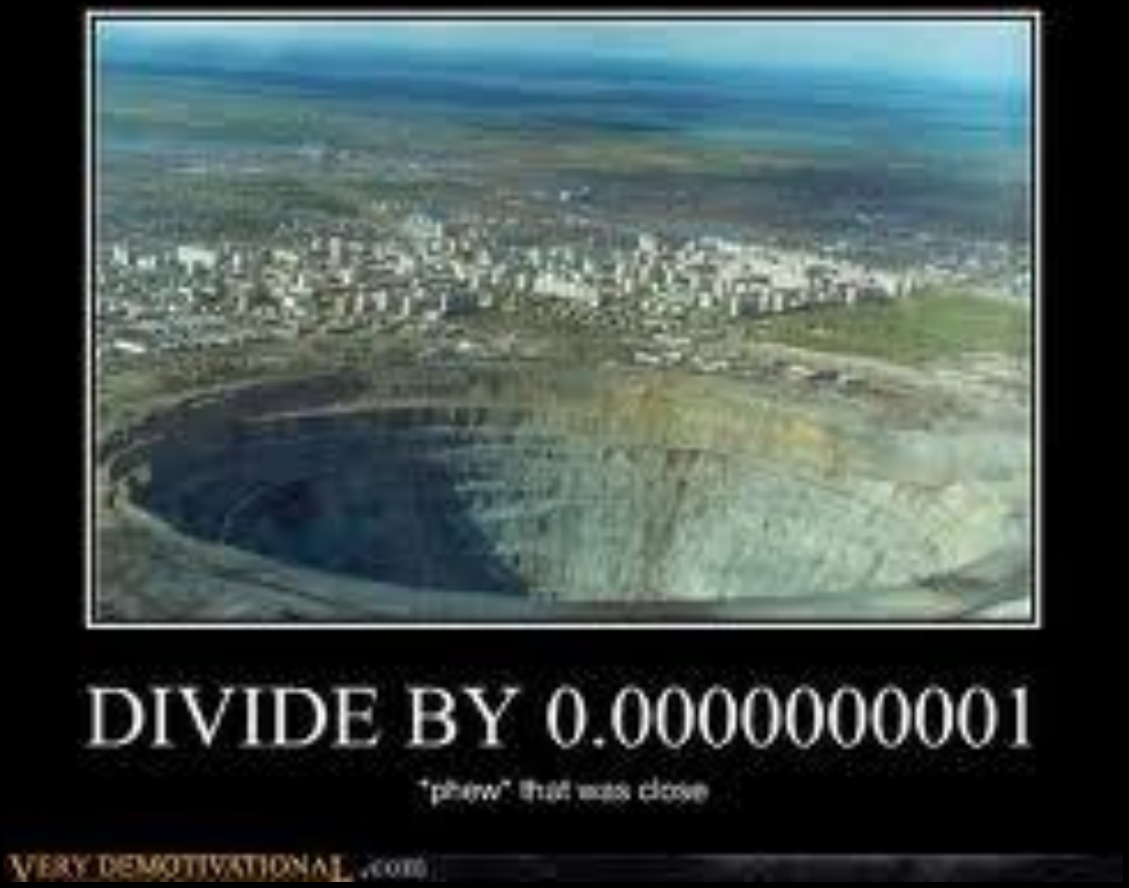
$$Y = H * F$$

$$F = \frac{Y}{H}$$

# Naïve Deconvolution



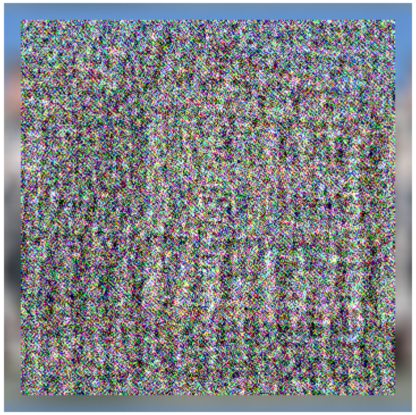
# Naïve Deconvolution



# Naïve Deconvolution

- PSF is a kind of low-pass filter
  - High-frequency magnitude is small (closed to zero)

$$B = K \otimes I$$

$$Y = H * F$$

$$F = \frac{Y}{H}$$

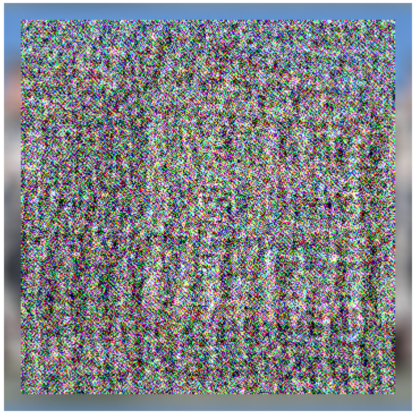
# Naïve Deconvolution

- Divide by zero.
  - *H* is close to zero
  - *F* could be extremely large, if Y is not zero. (Noise)

$$B = K \otimes I$$

$$Y = H * F$$

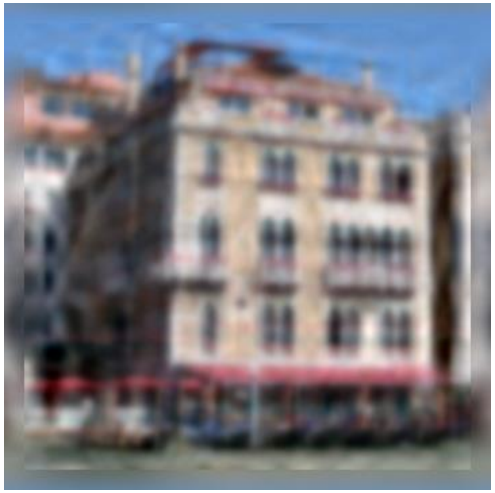
$$F = \frac{Y}{H}$$



# Wiener Deconvolution

- When $\lambda$ is large
  - Scalar term < 1
- When $\lambda$ is small
  - Scalar term ~= 1

$$F = \frac{Y}{H}\left(\underbrace{\frac{|H|^2}{|H|^2 + \lambda}}_{\text{Scalar term}}\right)$$



*λ = 10^-4*

*λ = 10^-7*

# Wiener Filter Derivation

Note that all * donate the complex conjugate

$$B = K \otimes I + N$$

$$Y = HF + V$$

$$\hat{F} = \frac{1}{H} Y = GY$$

$$\epsilon = E\left|F - \hat{F}\right|^2$$

$$= E|F - GY|^2$$

$$= E|F - G(HF + V)|^2$$

$$= E|F - GHF - GV|^2$$

$$= E|(1 - GH)F - GV|^2$$

$$= E\{\, [(1 - GH)F - GV]^*[(1 - GH)F - GV]\, \}$$

$$= E[(1 - GH)^* F^* (1 - GH)F - (1 - GH)^* F^* GV - G^* V^* (1 - GH)F + G^* V^* GV]$$

$$= E[(1 - GH)^* (1 - GH) F^* F - (1 - GH)^* G (VF^*) - (1 - GH) G^* (FV^*) + G^* G V^* V]$$

# Wiener Filter Derivation

Since we assume the noise is independent of the signal, therefore:

$$VF^* = FV^* = 0$$

$$\begin{aligned}\epsilon &= E[(1 - GH)^*(1 - GH)F^*F + G^*GV^*V] \\ &= E[(1 - GH)^*(1 - GH)|F|^2 + G^*G|V|^2] \\ &= E(|F|^2)(1 - GH)^*(1 - GH) + E(|V|^2)G^*G\end{aligned}$$

To find the minimum error value, we calculate the Wirtinger derivative with respect to $G$ and set it equal to zero.

$$\frac{d\epsilon}{dG} = E(|V|^2)G^* - E(|F|^2)H(1 - GH)^* = 0$$

$$E(|V|^2)G^* - E(|F|^2)H + E(|F|^2)G^*|H|^2 = 0$$

$$[E(|V|^2) + E(|F|^2)|H|^2]G^* - E(|F|^2)H = 0$$

$$G^* = \frac{E(|F|^2)H}{E(|V|^2) + E(|F|^2)|H|^2}$$

# Wiener Filter Derivation

$$G = \frac{E(|F|^2)H^*}{E(|V|^2) + E(|F|^2)|H|^2} = \frac{H^*}{|H|^2 + \frac{E(|V|^2)}{E(|F|^2)}} = \frac{1}{H}\frac{HH^*}{|H|^2 + \frac{E(|V|^2)}{E(|F|^2)}}$$

$$= \frac{1}{H}\frac{|H|^2}{|H|^2 + \frac{E(|V|^2)}{E(|F|^2)}} = \frac{1}{H}\frac{|H|^2}{|H|^2 + \lambda}$$

$$\hat{F} = \frac{1}{H}Y = GY = \frac{Y}{H}\frac{|H|^2}{|H|^2 + \lambda}$$

# How do we know noise to signal ratio?

- They are different in each frequency.

$$F_f = \frac{Y_f}{H_f}\left(\frac{\left|H_f\right|^2}{\left|H_f\right|^2 + \lambda_f}\right) \qquad \lambda_f = \frac{E[\left|V_f\right|^2]}{E[\left|F_f\right|^2]} = \frac{Noise}{Signal}$$

- Strategy
  - Small in low frequency
  - Large in high frequency

Richardson–Lucy deconvolution

# SPATIAL DECONVOLUTION

# Normal Equation

- Convolution in matrix multiplication form
  - *A*: convolution with *K* in matrix form
  - *x*: original image *i* reshape into one dimension
  - *b*: blurry image *B* reshape into one dimension

$$K \otimes I = B$$

$$\min_{x} \|Ax - b\|^2 \Rightarrow Ax = b$$

# Poisson distribution

- During exposure, the probability (times) of pixel intensity (*k*) to be observed (captured).

$$f(k,\lambda) = P(x = k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

$$E(x) = \lambda$$

# Richardson–Lucy deconvolution

- Poisson Distribution

$$P = f(k,\lambda) = \frac{\lambda^k e^{-\lambda}}{k!}$$

- Expectation image

$$\lambda = E(I) = K \otimes I = Ax$$

$$(Ax)_i = \sum_j A_{ij} x_j$$

- Observed image

$$k = b$$

# Richardson–Lucy deconvolution

- For each pixel *i*

$$P(b_i \mid x) = \frac{\lambda^k e^{-\lambda}}{k!} = \frac{(Ax)_i^{b_i} e^{-(Ax)_i}}{b_i!}$$

- For whole image

$$P(b \mid x) = \prod_i P(b_i \mid x)$$

# Richardson–Lucy deconvolution

- Maximum Log-likelihood

$$\arg\max_I \log(P(b\mid x)) = \log\left(\prod_i \frac{\lambda^k e^{-\lambda}}{k!}\right)$$

$$= \sum_i \left\{k \cdot \log(\lambda) - \lambda - \log(k!)\right\}$$

$$\propto \sum_i \left\{k \cdot \log(\lambda) - \lambda\right\}$$

$$= \sum_i \left\{b_i \cdot \log(Ax)_i - (Ax)_i\right\}$$

- Energy minimization

$$Energy(x) = \sum_i \left\{(Ax)_i - b_i \cdot \log(Ax)_i\right\}$$

# Richardson–Lucy deconvolution

- Energy minimization

$$Energy(x) = \sum_i \left\{ (Ax)_i - b_i \cdot \log(Ax)_i \right\}$$

- Take derivative to minimize energy

$$\frac{\partial Energy(x)}{\partial x} = \begin{bmatrix} \dfrac{\partial \sum_i \left\{ (Ax)_i - b_i \cdot \log(Ax)_i \right\}}{\partial x_1} \\ \dfrac{\partial \sum_i \left\{ (Ax)_i - b_i \cdot \log(Ax)_i \right\}}{\partial x_2} \\ \vdots \\ \dfrac{\partial \sum_i \left\{ (Ax)_i - b_i \cdot \log(Ax)_i \right\}}{\partial x_n} \end{bmatrix} = 0$$

# Richardson–Lucy deconvolution

$$\frac{\partial \sum_i \left\{ (Ax)_i - b_i \cdot \log(Ax)_i \right\}}{\partial x_l} = \sum_i \left[ \frac{\partial \left\{ (Ax)_i \right\}}{\partial x_l} - \frac{\partial \left\{ b_i \cdot \log(Ax)_i \right\}}{\partial x_l} \right]$$

$$= \sum_i \left[ \frac{\partial \sum_j A_{ij} x_j}{\partial x_l} - \frac{\partial b_i \cdot \log(\sum_j A_{ij} x_j)}{\partial x_l} \right] = \sum_i \left\{ A_{il} - \frac{b_i A_{il}}{\sum_j A_{ij} x_j} \right\}$$

$$\Rightarrow \sum_i A_{il} - \sum_i \frac{b_i A_{il}}{\sum_j A_{ij} x_j} = 0$$

# Richardson–Lucy deconvolution

$$\because \sum_i A_{il} = 1 \qquad \therefore \sum_i \frac{b_i A_{il}}{\sum_j A_{ij} x_j} = 1$$

Assuming the convergence ratio $\dfrac{x^{t+1}}{x^t} = 1$

$$x_l^{t+1} = x_l^t \left[ \sum_i \frac{b_i A_{il}}{\sum_j A_{ij} x_j^t} \right]$$

$$I^{t+1} = I^t \cdot \left[ K^* \otimes \frac{B}{I^t \otimes K} \right]$$

*K** donates flipped *K*
The division and multiplication are element wised.

# Richardson–Lucy deconvolution

- Additive form - Gradient based

$$I^{t+1} = I^{t} + \Delta\left[1 - K^{*} \otimes \frac{B}{I^{t} \otimes K}\right]$$

$\Delta$ is the gradient step.

Non-Blind Deconvolution

# OPTIMIZATION

# Normal Equation

- Convolution in matrix multiplication form
  - *A*: convolution with *K* in matrix form
  - *x*: original image *I* reshape into one dimension
  - *b*: blurry image *B* reshape into one dimension

$$K \otimes I = B$$

$$\min_{x} \|Ax - b\|^2 \Rightarrow Ax = b$$

# Normal Equation

The quadratic form

$$x^T A x + b^T x + c$$

Why $Ax = b \leftrightarrow arg\,min_x \|Ax - b\|^2$ ?

$$\|Ax - b\|^2 = (Ax - b)^T(Ax - b) = (x^T A^T - b^T)(Ax - b)$$
$$= x^T A^T A x - x^T A^T b - b^T A x + b^T b$$

Since $x^T A^T b$ is a scalar equal to $b^T A x$

$$x^T A^T A x - x^T A^T b - b^T A x + b^T b$$
$$= x^T A^T A x - 2x^T A^T b + b^T b$$
$$= x^T A^T A x - 2b^T A x + b^T b$$

Quadratic form!

Take first-order differential respect to $x^T$ equating to zero for minimization

$$\frac{d(x^T A^T A x - 2x^T A^T b + b^T b)}{d\,x^T} = 0$$
$$2A^T A x - 2A^T b = 0$$

If $A$ is not symmetric, solve ($A^T A$ is symmetric positive semi-definite)

$$A^T A x = A^T b$$

If $A$ is symmetric, solve ($A$ is symmetric positive definite)

$$Ax = b$$

# Spatial Convolution

- 3x3 image convolution with 3x3 box blur kernel example
  - Border constant zero

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |

$\otimes$

| | | |
|---|---|---|
| x11 | x12 | x13 |
| x21 | x22 | x23 |
| x31 | x32 | x33 |

# Spatial Convolution as a Linear System

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | | x11 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | | x12 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | | x13 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | | x21 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | x22 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | | x23 |
| 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | | x31 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | | x32 |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | | x33 |

| | | |
|---|---|---|
| x11 | x12 | x13 |
| x21 | x22 | x23 |
| x31 | x32 | x33 |

# Optimization

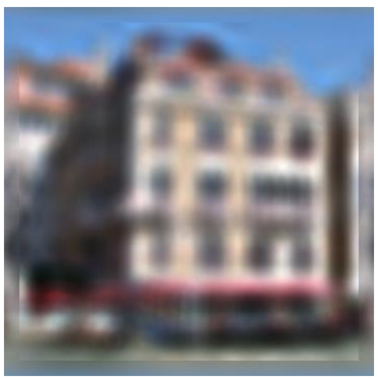
- Properties $Ax = b$
  - *A* is large and sparse (m x n) x (m x n)
  - ***Not possible*** **to *save* and *compute multiplication***
  - Use linear convolution $Ax = K \otimes I$

# Conjugate Gradient Method

- Why Conjugate Gradient ?
  - Converge *fast* with good initial guess and preconditioner
  - Replace matrix multiplication with convolution

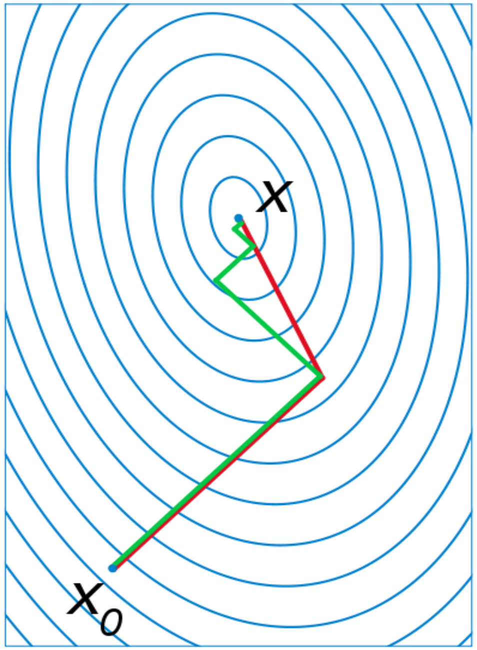
initial image

Conjugate Gradient



Richardson–Lucy

10 iterations




http://en.wikipedia.org/wiki/Conjugate_gradient_method

# Conjugate Gradient Method

– Bottleneck: $Ap_k$

– Note that $A$ is ***not*** symmetric

– Solve $A^T A x = A^T b$

$$Ax = b$$

$\mathbf{r}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0$
$\mathbf{p}_0 := \mathbf{r}_0$
$k := 0$
repeat
$\qquad \alpha_k := \frac{\mathbf{r}_k^{\mathrm{T}}\mathbf{r}_k}{\mathbf{p}_k^{\mathrm{T}}\mathbf{A}\mathbf{p}_k}$
$\qquad \mathbf{x}_{k+1} := \mathbf{x}_k + \alpha_k \mathbf{p}_k$
$\qquad \mathbf{r}_{k+1} := \mathbf{r}_k - \alpha_k \mathbf{A}\mathbf{p}_k$
$\qquad$ if $r_{k+1}$ is sufficiently small then exit loop
$\qquad \beta_k := \frac{\mathbf{r}_{k+1}^{\mathrm{T}}\mathbf{r}_{k+1}}{\mathbf{r}_k^{\mathrm{T}}\mathbf{r}_k}$
$\qquad \mathbf{p}_{k+1} := \mathbf{r}_{k+1} + \beta_k \mathbf{p}_k$
$\qquad k := k + 1$
end repeat

http://en.wikipedia.org/wiki/Conjugate_gradient_method

# Preconditioned Conjugate Gradient

- Faster convergence.
- Overhead
  - Multiply

    Preconditioner in loop
  - Find good preconditioner
    - incomplete Cholesky factorization
    - Cheap for multiplication

$$\mathbf{r}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0$$
$$\mathbf{z}_0 := \mathbf{M}^{-1}\mathbf{r}_0$$
$$\mathbf{p}_0 := \mathbf{z}_0$$
$$k := 0$$

**repeat**

$$\alpha_k := \frac{\mathbf{r}_k^{\mathrm{T}}\mathbf{z}_k}{\mathbf{p}_k^{\mathrm{T}}\mathbf{A}\mathbf{p}_k}$$
$$\mathbf{x}_{k+1} := \mathbf{x}_k + \alpha_k\mathbf{p}_k$$
$$\mathbf{r}_{k+1} := \mathbf{r}_k - \alpha_k\mathbf{A}\mathbf{p}_k$$

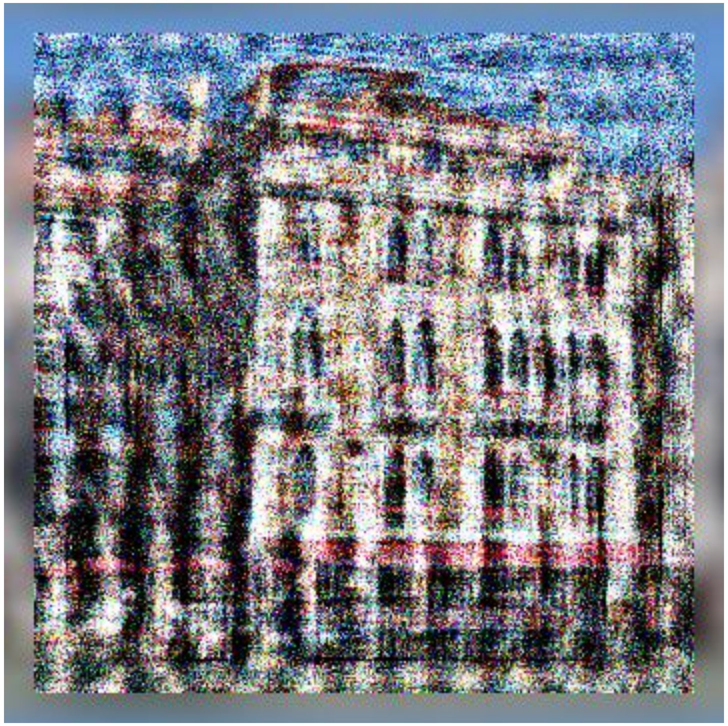
**if** $r_{k+1}$ is sufficiently small **then** exit loop **end if**

$$\mathbf{z}_{k+1} := \mathbf{M}^{-1}\mathbf{r}_{k+1}$$
$$\beta_k := \frac{\mathbf{z}_{k+1}^{\mathrm{T}}\mathbf{r}_{k+1}}{\mathbf{z}_k^{\mathrm{T}}\mathbf{r}_k}$$
$$\mathbf{p}_{k+1} := \mathbf{z}_{k+1} + \beta_k\mathbf{p}_k$$
$$k := k + 1$$

**end repeat**

# Over Convergence

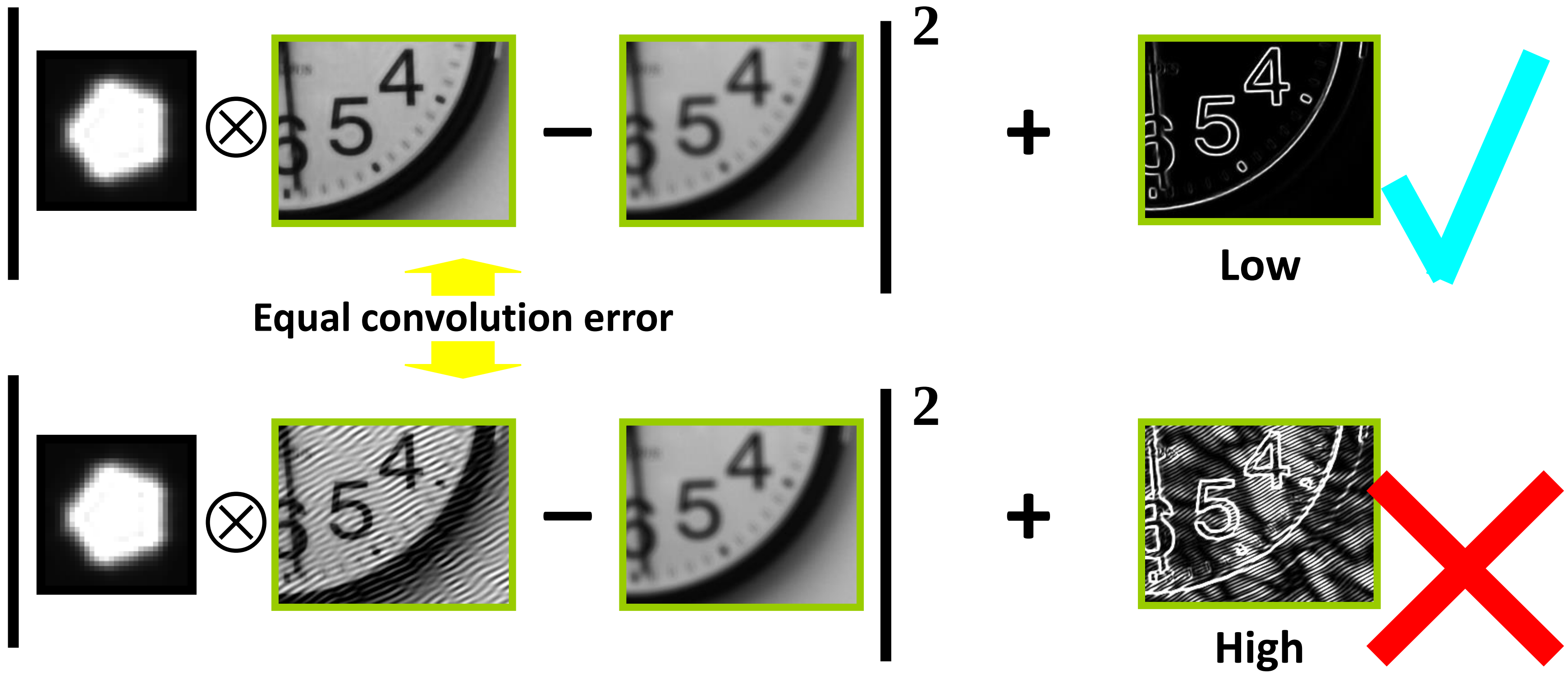
10 iterations



200 iterations

# Put Penalty on Ringing

- ill-posed problem




*Deconvolution using natural image priors SiGGRAPH 2007, Levin et al.*

# Put Penalty on Ringing

- Smooth regions should not appear ringing.
  - Preserve smooth region from blurry image.
  - Only reconstruct the gradient region.
- Drawback
  - Failed when PSF is large ( >30 pixels )
    - Gradient magnitude in blurry image is low.
    - Ringing in overlap region.
  - Lack of high frequency detail.

(a)

(c)

*High-quality Non-blind image Deconvolution PSiVT 2010, Jong-Ho et al.*

# Regularization

$$\min_x \|Ax - b\|^2 + \boxed{\lambda \|\nabla x\|^\alpha}$$

Regularization/Penalty term

$\|\nabla x\|^2$

**“spread” gradients**

$\|\nabla x\|^{0.8}$

**“localizes” gradients**

**Richardson-Lucy**

**Gaussian prior**

**Sparse prior**

*Deconvolution using natural image priors SiGGRAPH 2007, Levin et al.*

# Regularization

$$\min_x \|Ax - b\|^2 + \lambda \|\nabla x\|^\alpha$$

$\alpha = \infty$ $\alpha = 2$ $\alpha = 1$ $0 < \alpha < 1$ $\alpha = 0$

# Regularization

- Properties
  - L2-Norm
    - force the coefficients to be more similar to each other in order to minimize their joint 2-norm
    - Easy to compute
  - L1-Norm
    - L2 + sparse
    - Hard to solve
  - Want sparse result, apply L1 norm.

# Regularization $\alpha = 2$

$$arg\,\min_x \|Ax - b\|^2 + \lambda\|\Gamma x\|^2 = x^T A^T Ax - 2x^T A^T b + b^T b + \lambda x^T \Gamma^T \Gamma x$$

Take first-order differential respect to $x^T$ equating to zero for minimization

$$\frac{d(x^T A^T Ax - 2x^T A^T b + b^T b + \lambda x^T \Gamma^T \Gamma x)}{d\,x^T} = 0$$

$$2A^T Ax - 2A^T b + 2\lambda\Gamma^T \Gamma x = 0$$

$$A^T Ax + \lambda\Gamma^T \Gamma x = A^T b$$

$$(A^T A + \lambda\Gamma^T \Gamma)x = A^T b$$

$$x = (A^T A + \lambda\Gamma^T \Gamma)^{-1} A^T b$$

# Regularization $\alpha = 2$

- Tikhonov regularization
  - L2-Norm regularization

$$\min_x \|Ax - b\|^2 + \lambda \|\Gamma x\|^2$$

$$(A^T A + \lambda \Gamma^T \Gamma) x = A^T b$$

$$x = (A^T A + \lambda \Gamma^T \Gamma)^{-1} A^T b$$

# Regularization $\alpha = 2$

- Tikhonov regularization
  - Solve in frequency domain

$$x = (A^T A + \lambda \Gamma^T \Gamma)^{-1} A^T b$$

$$= \frac{A^T b}{A^T A + \lambda \Gamma^T \Gamma}$$

$$X(u,v) = \frac{F(u,v)^* Y(u,v)}{F(u,v)^* F(u,v) + \lambda G(u,v)^* G(u,v)}$$

$$= \frac{F(u,v)^* Y(u,v)}{\left|F(u,v)\right|^2 + \lambda \left|G(u,v)\right|^2}$$

* donates the complex conjugate

=> donates the Fast Fourier Transform

$$A(x,y) \Rightarrow F(u,v)$$

$$A^T(x,y) \Rightarrow F^*(u,v)$$

$$\Gamma(x,y) \Rightarrow G(u,v)$$

$$\Gamma^T(x,y) \Rightarrow G^*(u,v)$$

$$b(x,y) \Rightarrow Y(u,v)$$

$$QQ^* = \left|Q\right|^2$$

# Regularization $\alpha = 2$

- Tikhonov regularization
  - Solve in frequency domain
  - Trade-off
    - Fast!! O(N logN) direct method
    - Some inaccuracies at the image boundaries

# Regularization $\alpha \neq 2$

- Iterative Shrinkage-Thresholding Algorithm (ISTA) $\alpha = 1$
- Truncated Newton interior-point method
  $\alpha = 1$
- Iterative re-weighted least squares (IRLS)
  $\alpha < 2$

$$\min_x \left\| Ax - b \right\|^2 + \lambda \left\| \nabla x \right\|^\alpha$$

# PSF anchor problem

- Anchor usually at the center of PSF
  - Circular shift before FFT
  - Matlab function: psf2otf


# Boundary Artifacts

- Lost of boundary intensity.
- Periodicity of the data. (FFT based convolution)

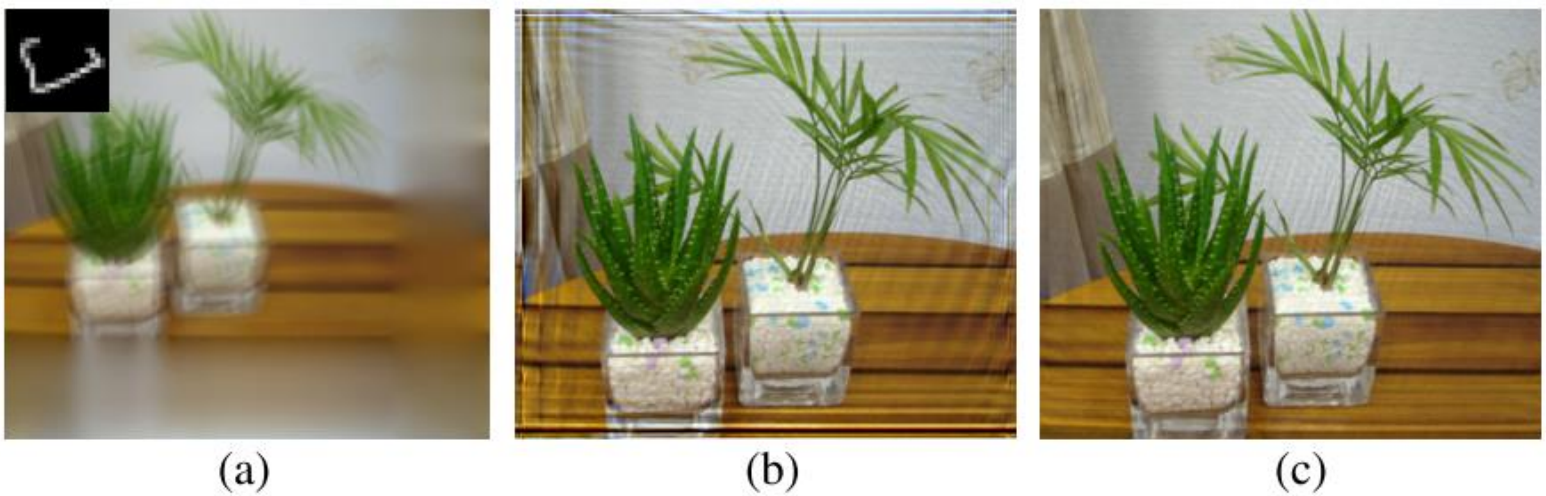



**Fig. 3.** Reducing boundary artifacts. (a) Expanded blurred image. (b) Richardson-Lucy result without RBA algorithm. (c) Richardson-Lucy result with RBA algorithm.

*Non-blind Image Deconvolution with Adaptive Regularization, Jong-Ho Lee et al. PCM 2010*

Blind Deconvolution

# OPTIMIZATION

# Algorithm

- Estimate point spread function
- Estimate latent image
  - Non-blind

---

**Algorithm 1** : Overall Algorithm

---

**Require:** Observed blurry image $g$, Maximum kernel size $h$.

Apply derivative filters to $g$, creating a high-freq. image $y$.

1. Blind estimation of blur matrix $K$ (Section 3.1) from $y$.

   Loop over coarse-to-fine levels:

   Alternate:

   - Update sharp high-frequency image $x$ (Section 3.1.1) using $l_1/l_2$ regularization.
   - Update blurring matrix $K$ (Section 3.1.2).

   Interpolate solution to finer level as initialization.

2.Image recovery using non-blind algorithm of [12] (Section 3.2).

   - Deblur $g$ using $K$ to give sharp image $u$.

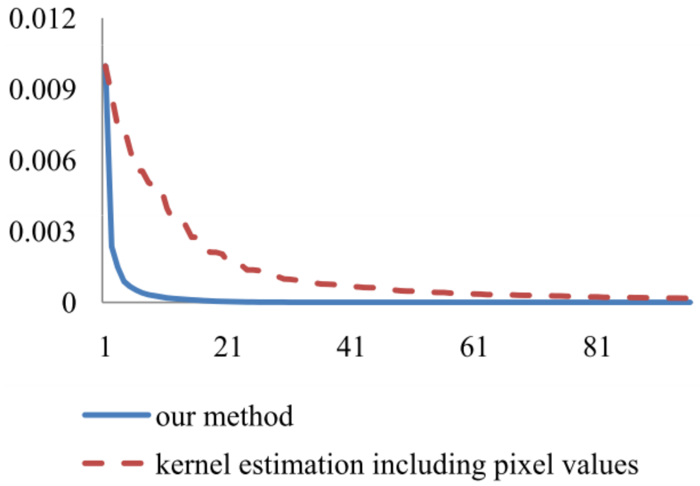
**return** Sharp image $u$.

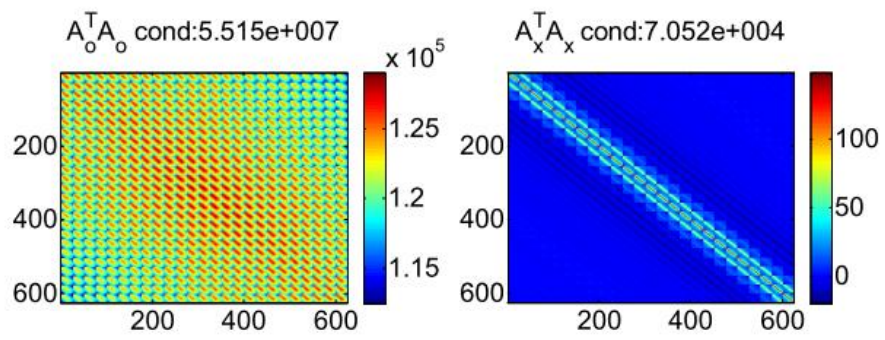
---

*Blind Deconvolution Using a Normalized Sparsity Measure CVPR 2011, Krishnan et al.*

# Estimate point spread function

- In gradient domain
  - Avoid boundary artifact (due to sparsity)
  - Faster convergence (low condition number)

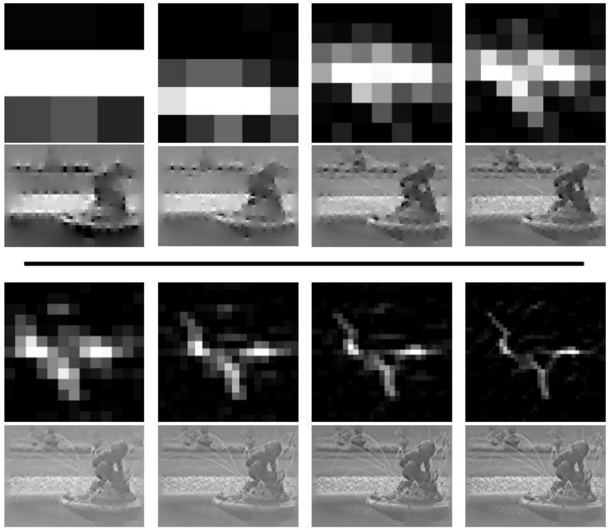



*Fast Motion Deblurring, Sunghyun Cho et al., SIGGRAPH 2009*

# Kernel(PSF) Estimation

- Multi-scale / Pyramid Approach
  - For large point spread function (> 30x30)
  - Avoid trapped into local minimal
  - Corse result as the finer initial



Removing camera shake from a single photograph SiGGRAPH 2006, Fergus et al.

# Kernel Estimation

- L-2 norm regularization
  - Easy to compute
  - Need suppression in low intensity element (kernel noise)
  - Canny edge
- L-1 norm regularization
  - Hard to compute
  - Great sparse properties.

*Image Deblurring with BlurredNoisy image Pairs SiGGRAPH 2007*

# Spatial Convolution

- 3x3 image convolution with 3x3 unknown kernel
  - Border zero

| | | | |
|---|---|---|---|
| | | | |
| | i11 | i12 | i13 |
| | i21 | i22 | i23 |
| | i31 | i32 | i33 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | i11 | i12 | 0 | i21 | i22 |
| 0 | 0 | 0 | i11 | i12 | i13 | i21 | i22 | i23 |
| 0 | 0 | 0 | i12 | i13 | 0 | i22 | i23 | 0 |
| 0 | i11 | i12 | 0 | i21 | i22 | 0 | i31 | i32 |
| i11 | i12 | i13 | i21 | i22 | i23 | i31 | i32 | i33 |
| i12 | i13 | 0 | i22 | i23 | 0 | i32 | i33 | 0 |
| 0 | i21 | i22 | 0 | i32 | i32 | 0 | 0 | 0 |
| i21 | i22 | i23 | i31 | i32 | i33 | 0 | 0 | 0 |
| i22 | i23 | 0 | i32 | i33 | 0 | 0 | 0 | 0 |

# Spatial Convolution

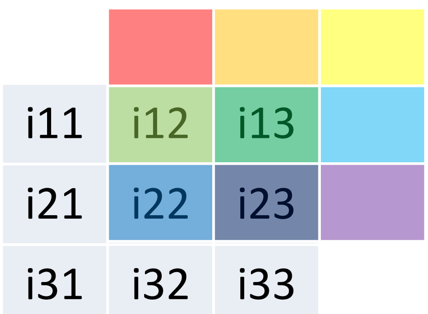
- 3x3 image convolution with 3x3 unknown kernel
  - Border zero

| | | |
|---|---|---|
| | | |
| i11 | i12 | i13 |
| i21 | i22 | i23 |
| i31 | i32 | i33 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | i11 | i12 | 0 | i21 | i22 |
| 0 | 0 | 0 | i11 | i12 | i13 | i21 | i22 | i23 |
| 0 | 0 | 0 | i12 | i13 | 0 | i22 | i23 | 0 |
| 0 | i11 | i12 | 0 | i21 | i22 | 0 | i31 | i32 |
| i11 | i12 | i13 | i21 | i22 | i23 | i31 | i32 | i33 |
| i12 | i13 | 0 | i22 | i23 | 0 | i32 | i33 | 0 |
| 0 | i21 | i22 | 0 | i32 | i32 | 0 | 0 | 0 |
| i21 | i22 | i23 | i31 | i32 | i33 | 0 | 0 | 0 |
| i22 | i23 | 0 | i32 | i33 | 0 | 0 | 0 | 0 |

# Spatial Convolution

- 3x3 image convolution with 3x3 unknown kernel
  - Border zero




| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | i11 | i12 | 0 | i21 | i22 |
| 0 | 0 | 0 | i11 | i12 | i13 | i21 | i22 | i23 |
| 0 | 0 | 0 | i12 | i13 | 0 | i22 | i23 | 0 |
| 0 | i11 | i12 | 0 | i21 | i22 | 0 | i31 | i32 |
| i11 | i12 | i13 | i21 | i22 | i23 | i31 | i32 | i33 |
| i12 | i13 | 0 | i22 | i23 | 0 | i32 | i33 | 0 |
| 0 | i21 | i22 | 0 | i32 | i32 | 0 | 0 | 0 |
| i21 | i22 | i23 | i31 | i32 | i33 | 0 | 0 | 0 |
| i22 | i23 | 0 | i32 | i33 | 0 | 0 | 0 | 0 |

# Spatial Convolution

- 3x3 image convolution with 3x3 unknown kernel
  - Border zero

| | | | |
|---|---|---|---|
| | i11 | i12 | i13 |
| | i21 | i22 | i23 |
| | i31 | i32 | i33 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | i11 | i12 | 0 | i21 | i22 |
| 0 | 0 | 0 | i11 | i12 | i13 | i21 | i22 | i23 |
| 0 | 0 | 0 | i12 | i13 | 0 | i22 | i23 | 0 |
| 0 | i11 | i12 | 0 | i21 | i22 | 0 | i31 | i32 |
| i11 | i12 | i13 | i21 | i22 | i23 | i31 | i32 | i33 |
| i12 | i13 | 0 | i22 | i23 | 0 | i32 | i33 | 0 |
| 0 | i21 | i22 | 0 | i32 | i32 | 0 | 0 | 0 |
| i21 | i22 | i23 | i31 | i32 | i33 | 0 | 0 | 0 |
| i22 | i23 | 0 | i32 | i33 | 0 | 0 | 0 | 0 |

# Spatial Convolution

- 3x3 image convolution with 3x3 unknown kernel
  - Border zero

| | | |
|---|---|---|
| i11 | i12 | i13 |
| i21 | i22 | i23 |
| i31 | i32 | i33 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | i11 | i12 | 0 | i21 | i22 |
| 0 | 0 | 0 | i11 | i12 | i13 | i21 | i22 | i23 |
| 0 | 0 | 0 | i12 | i13 | 0 | i22 | i23 | 0 |
| 0 | i11 | i12 | 0 | i21 | i22 | 0 | i31 | i32 |
| i11 | i12 | i13 | i21 | i22 | i23 | i31 | i32 | i33 |
| i12 | i13 | 0 | i22 | i23 | 0 | i32 | i33 | 0 |
| 0 | i21 | i22 | 0 | i32 | i32 | 0 | 0 | 0 |
| i21 | i22 | i23 | i31 | i32 | i33 | 0 | 0 | 0 |
| i22 | i23 | 0 | i32 | i33 | 0 | 0 | 0 | 0 |

# Spatial Convolution

- 3x3 image convolution with 3x3 unknown kernel
  - Border zero

| | | | |
|---|---|---|---|
| i11 | i12 | i13 | |
| i21 | i22 | i23 | |
| i31 | i32 | i33 | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | i11 | i12 | 0 | i21 | i22 |
| 0 | 0 | 0 | i11 | i12 | i13 | i21 | i22 | i23 |
| 0 | 0 | 0 | i12 | i13 | 0 | i22 | i23 | 0 |
| 0 | i11 | i12 | 0 | i21 | i22 | 0 | i31 | i32 |
| i11 | i12 | i13 | i21 | i22 | i23 | i31 | i32 | i33 |
| i12 | i13 | 0 | i22 | i23 | 0 | i32 | i33 | 0 |
| 0 | i21 | i22 | 0 | i32 | i32 | 0 | 0 | 0 |
| i21 | i22 | i23 | i31 | i32 | i33 | 0 | 0 | 0 |
| i22 | i23 | 0 | i32 | i33 | 0 | 0 | 0 | 0 |

# Spatial Convolution

- 3x3 image convolution with 3x3 unknown kernel
  - Border zero

| | | | |
|---|---|---|---|
| | i11 | i12 | i13 |
| | i21 | i22 | i23 |
| | i31 | i32 | i33 |
| | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | i11 | i12 | 0 | i21 | i22 |
| 0 | 0 | 0 | i11 | i12 | i13 | i21 | i22 | i23 |
| 0 | 0 | 0 | i12 | i13 | 0 | i22 | i23 | 0 |
| 0 | i11 | i12 | 0 | i21 | i22 | 0 | i31 | i32 |
| i11 | i12 | i13 | i21 | i22 | i23 | i31 | i32 | i33 |
| i12 | i13 | 0 | i22 | i23 | 0 | i32 | i33 | 0 |
| 0 | i21 | i22 | 0 | i32 | i32 | 0 | 0 | 0 |
| i21 | i22 | i23 | i31 | i32 | i33 | 0 | 0 | 0 |
| i22 | i23 | 0 | i32 | i33 | 0 | 0 | 0 | 0 |

# Spatial Convolution

- 3x3 image convolution with 3x3 unknown kernel
  - Border zero

| | | |
|---|---|---|
| i11 | i12 | i13 |
| i21 | i22 | i23 |
| i31 | i32 | i33 |
| | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | i11 | i12 | 0 | i21 | i22 |
| 0 | 0 | 0 | i11 | i12 | i13 | i21 | i22 | i23 |
| 0 | 0 | 0 | i12 | i13 | 0 | i22 | i23 | 0 |
| 0 | i11 | i12 | 0 | i21 | i22 | 0 | i31 | i32 |
| i11 | i12 | i13 | i21 | i22 | i23 | i31 | i32 | i33 |
| i12 | i13 | 0 | i22 | i23 | 0 | i32 | i33 | 0 |
| 0 | i21 | i22 | 0 | i32 | i32 | 0 | 0 | 0 |
| i21 | i22 | i23 | i31 | i32 | i33 | 0 | 0 | 0 |
| i22 | i23 | 0 | i32 | i33 | 0 | 0 | 0 | 0 |

# Spatial Convolution

- 3x3 image convolution with 3x3 unknown kernel
  - Border zero

| | | | |
|---|---|---|---|
| i11 | i12 | i13 | |
| i21 | i22 | i23 | |
| i31 | i32 | i33 | |
| | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | i11 | i12 | 0 | i21 | i22 |
| 0 | 0 | 0 | i11 | i12 | i13 | i21 | i22 | i23 |
| 0 | 0 | 0 | i12 | i13 | 0 | i22 | i23 | 0 |
| 0 | i11 | i12 | 0 | i21 | i22 | 0 | i31 | i32 |
| i11 | i12 | i13 | i21 | i22 | i23 | i31 | i32 | i33 |
| i12 | i13 | 0 | i22 | i23 | 0 | i32 | i33 | 0 |
| 0 | i21 | i22 | 0 | i32 | i32 | 0 | 0 | 0 |
| i21 | i22 | i23 | i31 | i32 | i33 | 0 | 0 | 0 |
| i22 | i23 | 0 | i32 | i33 | 0 | 0 | 0 | 0 |

# Spatial Convolution

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | i11 | i12 | 0 | i21 | i22 | | k11 | | b11 |
| 0 | 0 | 0 | i11 | i12 | i13 | i21 | i22 | i23 | | k12 | | b12 |
| 0 | 0 | 0 | i12 | i13 | 0 | i22 | i23 | 0 | | k13 | | b13 |
| 0 | i11 | i12 | 0 | i21 | i22 | 0 | i31 | i32 | | k21 | | b21 |
| i11 | i12 | i13 | i21 | i22 | i23 | i31 | i32 | i33 | x | k22 | = | b22 |
| i12 | i13 | 0 | i22 | i23 | 0 | i32 | i33 | 0 | | k23 | | b23 |
| 0 | i21 | i22 | 0 | i32 | i32 | 0 | 0 | 0 | | k31 | | b31 |
| i21 | i22 | i23 | i31 | i32 | i33 | 0 | 0 | 0 | | k32 | | b32 |
| i22 | i23 | 0 | i32 | i33 | 0 | 0 | 0 | 0 | | k33 | | b33 |

$$Ak = b$$

# Kernel Estimation (L-2 norm)

- Known
  - A is from latent (de-blurred) derivative image.
  - b is blurred image.
- Unknown
  - k is kernel.
  - Constrain sparsity of k

$$Ak = b$$

$$\min_k \|Ak - b\|^2 + \lambda \|k\|^2$$

# Kernel Estimation (L-1 norm)

$$Ak = b$$

$$\min_k \|Ak - b\|^2 + \lambda \|k\|_1$$

# Kernel Estimation (L-2 norm)

- Stack different derivative order/direction equations

$$Ak = b$$

$$\begin{bmatrix} A_x \\ A_y \\ A_{xx} \\ A_{yy} \\ A_{xy} \end{bmatrix} k = \begin{bmatrix} b_x \\ b_y \\ b_{xx} \\ b_{yy} \\ b_{xy} \end{bmatrix}$$

$$A\gamma \leftarrow \frac{\partial L}{\partial \gamma}$$

$$b\gamma \leftarrow \frac{\partial b}{\partial \gamma}$$

*Fast Motion Deblurring, Sunghyun Cho et al., SIGGRAPH 2009*

# Kernel Estimation (L-2 norm)

$$\min_k \|Ak - b\|^2 + \lambda \|k\|^2$$

$$\left(A^T A + \lambda I\right) k = A^T b$$

- Pre-compute

$$A^T A$$

$$A^T b$$

# Kernel Estimation (L-2 norm)

- Pre-compute acceleration

$$A^T A k = A^T b$$

$$\begin{bmatrix} A_x^T & A_y^T & A_{xx}^T & A_{yy}^T & A_{xy}^T \end{bmatrix} \begin{bmatrix} A_x \\ A_y \\ A_{xx} \\ A_{yy} \\ A_{xy} \end{bmatrix} k = \begin{bmatrix} A_x^T & A_y^T & A_{xx}^T & A_{yy}^T & A_{xy}^T \end{bmatrix} \begin{bmatrix} b_x \\ b_y \\ b_{xx} \\ b_{yy} \\ b_{xy} \end{bmatrix}$$

$$\left(A_x^T A_x + A_y^T A_y + A_{xx}^T A_{xx} + A_{yy}^T A_{yy} + A_{xy}^T A_{xy}\right) k = \left(A_x^T b + A_y^T b + A_{xx}^T b + A_{yy}^T b + A_{xy}^T b\right)$$

$$\left(\sum_{i \in \Omega} A_i^T A_i\right) k = \left(\sum_{i \in \Omega} A_i^T b_i\right), \Omega = \{i \mid x, y, xx, yy, xy\}$$

# Kernel Estimation (L-2 norm)

- Sparse Problem
  - m>1000000
  - 9<n<100

$$\min_{k} \|Ak - b\|^2 + \lambda \|k\|^2$$

$$\left( \underset{n\times m}{A^T} \underset{m\times n}{A} + \lambda I \right) \underset{n\times 1}{k} = \underset{n\times m}{A^T} \underset{m\times 1}{b}$$

- Multiply as convolution

$$\left( L^T \otimes L + \lambda \right) \otimes K = L^T \otimes B$$

# Convolution Valid Region


$Ak$

# Convolution Valid Region


$Ak$

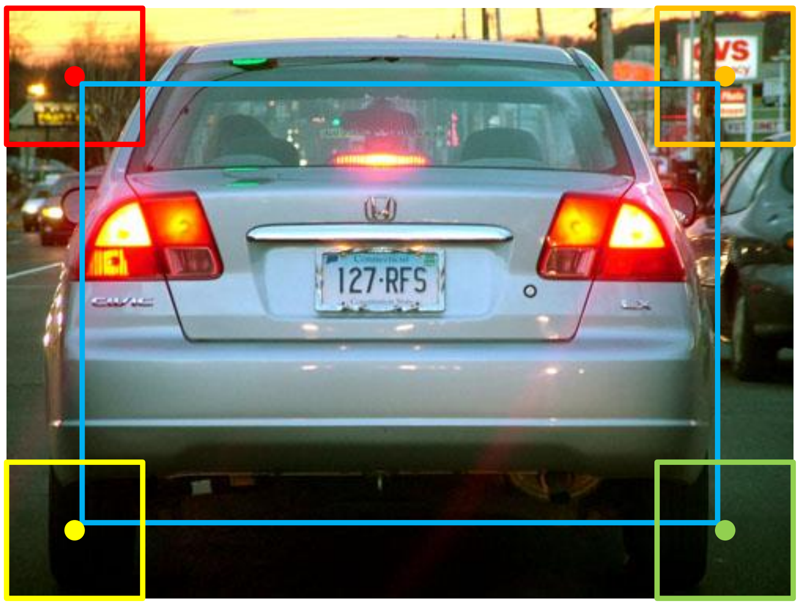
# Convolution Valid Region


$Ak$

# Convolution Valid Region


$Ak$

# Convolution Valid Region




$Ak$

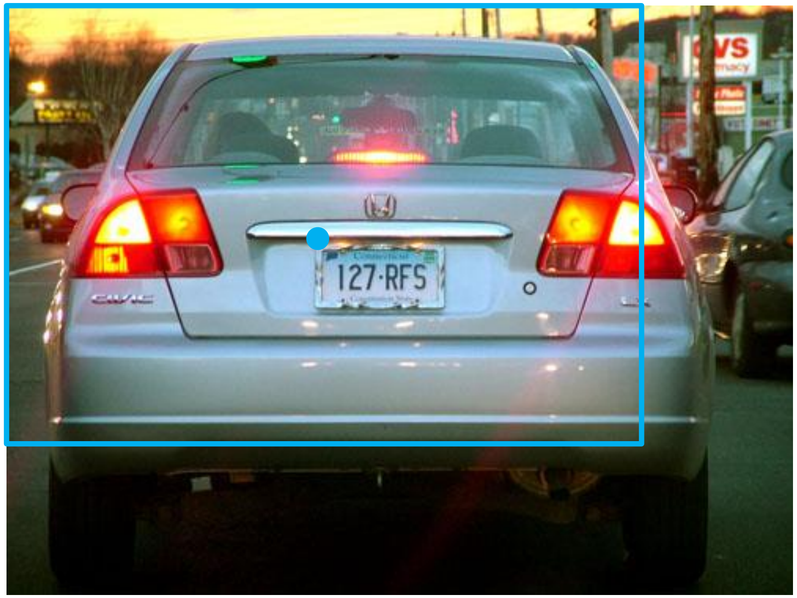
# Convolution Valid Region


$Ak$

# Convolution Valid Region

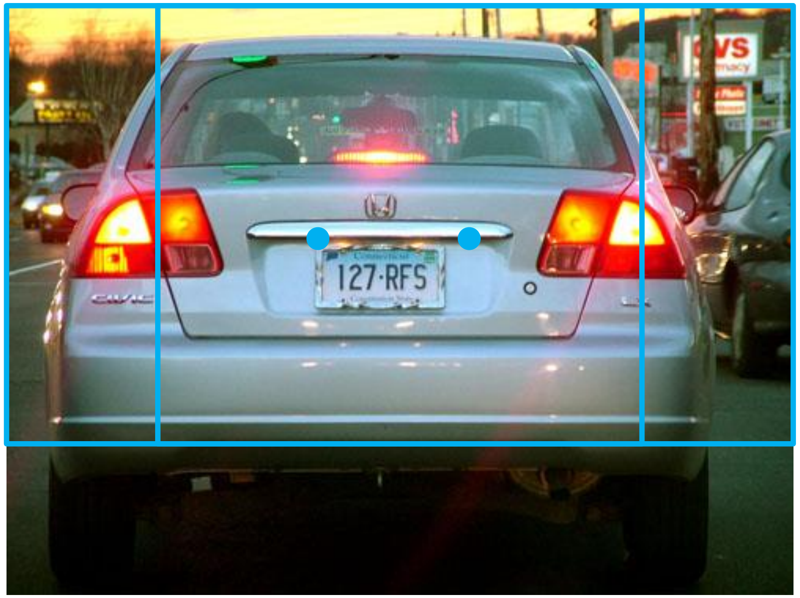

$Ak$

# Convolution Valid Region



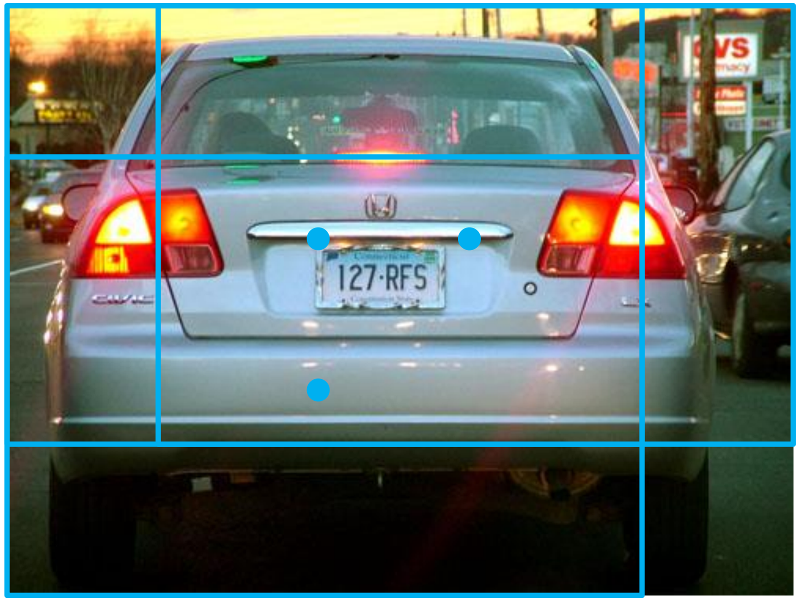

$A^T A k$

# Convolution Valid Region




$$A^T Ak$$

# Convolution Valid Region

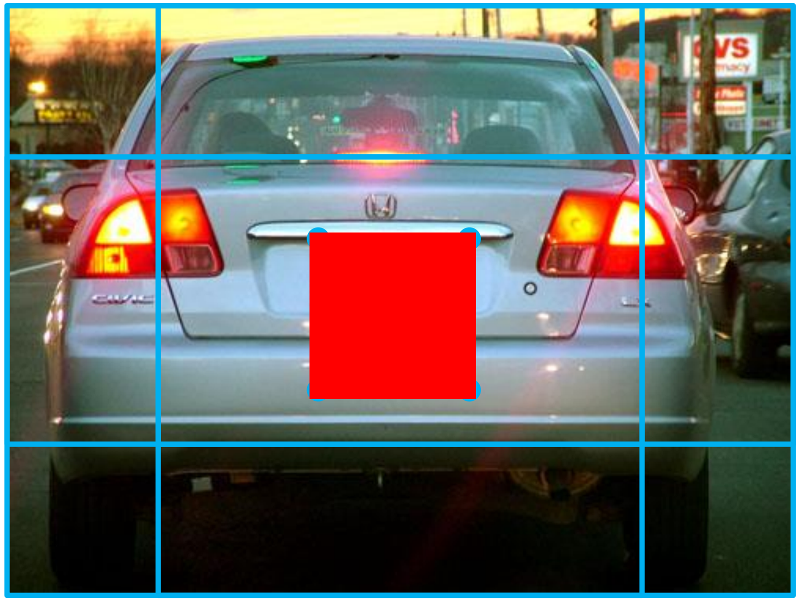



$$A^T A k$$

# Convolution Valid Region

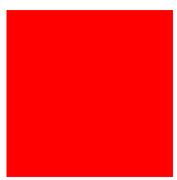

$$A^T A k$$

# Convolution Valid Region




$A^T A k$

# Convolution Valid Region



Same size of *k*

$A^T A k$

# EM-Algorithm

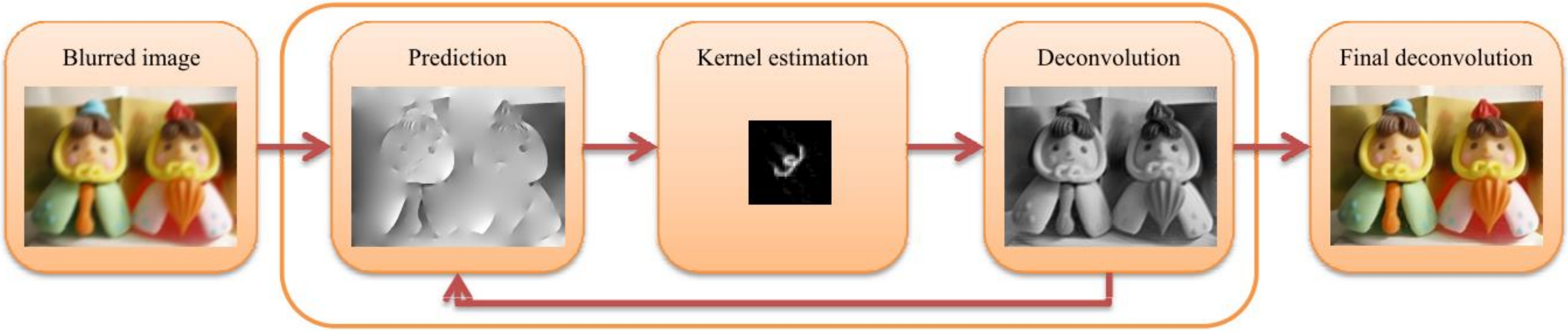
1. Estimate de-blurred image from given kernel.
2. Estimate kernel from given de-blurred image.
   – Normalize kernel value
3. Interpolate to next resolution level

$$\min_{x} \|Kx - b\|^2 + \lambda_1 \|x\|^2$$

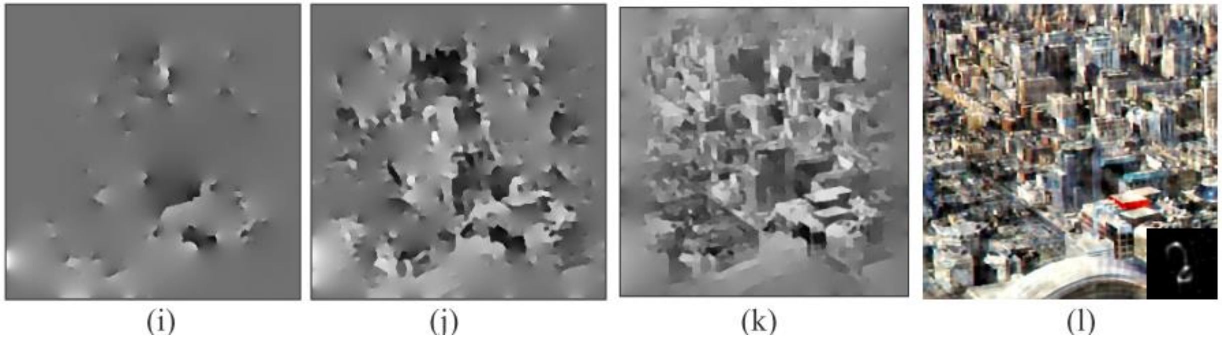
$$\min_{k} \|Ak - b\|^2 + \lambda_2 \|k\|^2$$

# Feature Extraction

- Estimate kernel with good edge feature
  - Processing strong edges




*Fast Motion Deblurring, Sunghyun Cho et al., SIGGRAPH 2009*




*Two-Phase Kernel Estimation for Robust Motion Deblurring. Li Xu et al.*

# Feature Extraction

- Bilateral filter
- Shock filter
- Gradient magnitude threshold

# Feature Extraction

- Strong edges can damage kernel estimation
  - Small object edges (ex: point light)

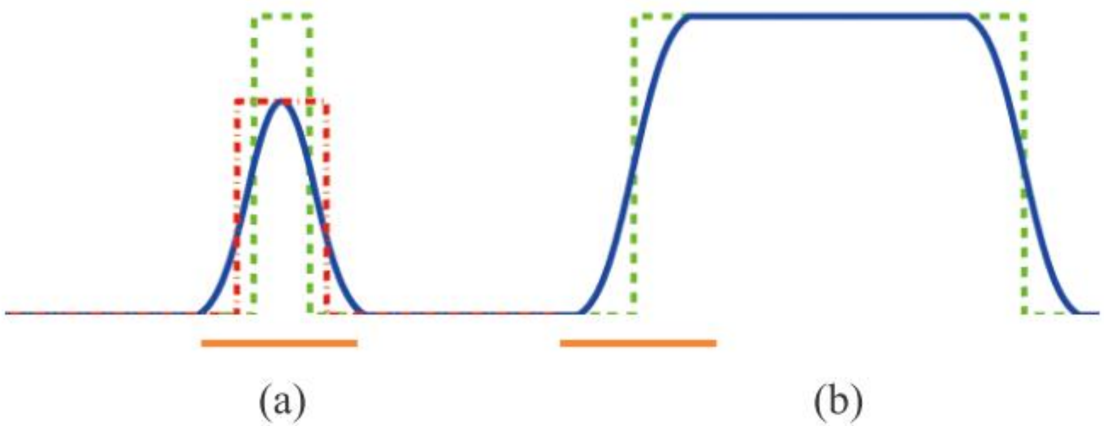


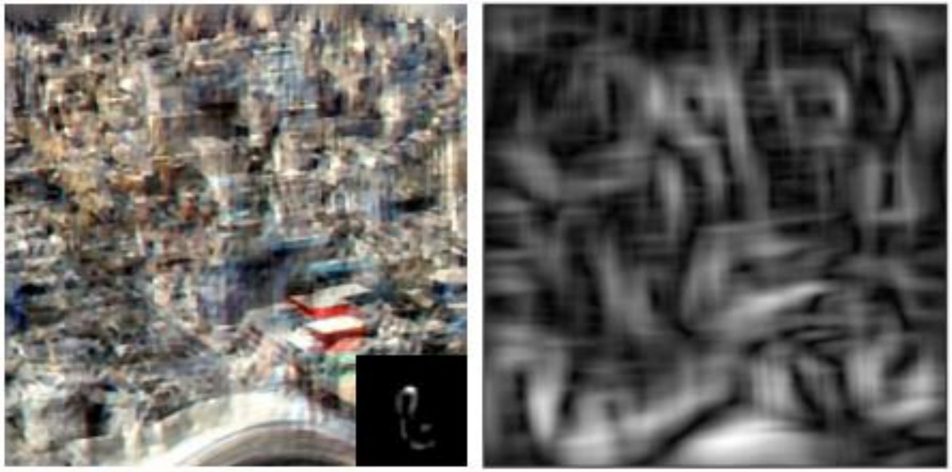

**Fig. 1.** Ambiguity in motion deblurring. Two latent signals (green dashed lines) in (a) and (b) are blurred (shown in blue) with the same Gaussian kernel. In (a), the blurred signal is not total-variation preserving, making the kernel estimation ambiguous. In fact, the red curve is more likely the latent signal than the green one in a common optimization process. The bottom orange lines indicate the input kernel width.

*Two-Phase Kernel Estimation for Robust Motion Deblurring. Li Xu et al.*

# Feature Extraction

- Eliminate small object edges.

$$r(x) = \frac{\boxed{\|\sum_{y \in N_h(x)} \nabla B(y)\|}}{\sum_{y \in N_h(x)} \|\nabla B(y)\| + 0.5}$$



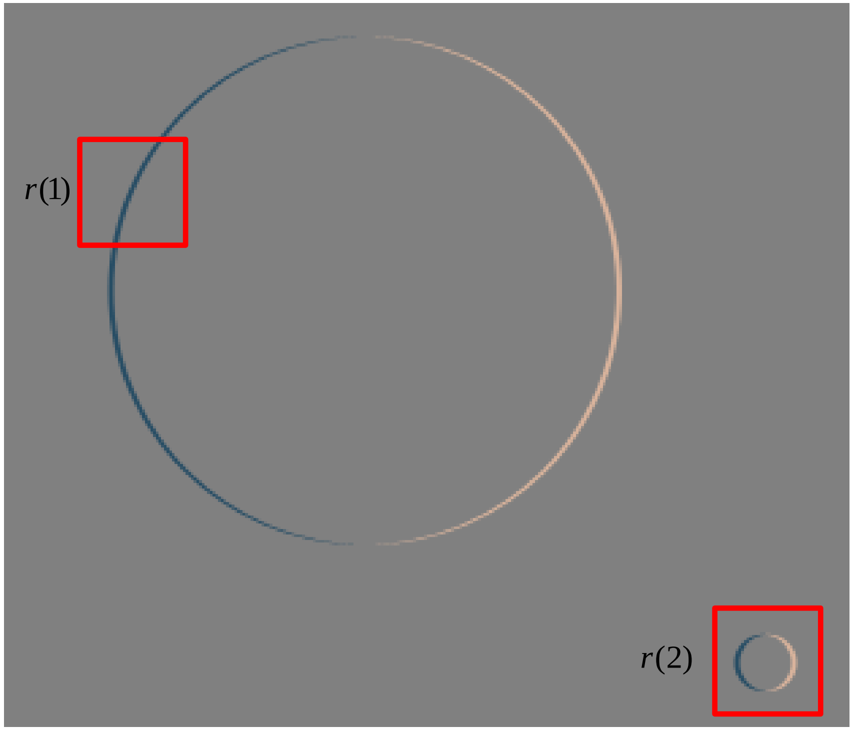
*Two-Phase Kernel Estimation for Robust Motion Deblurring. Li Xu et al.*

# Feature Extraction

- Eliminate small object edges.

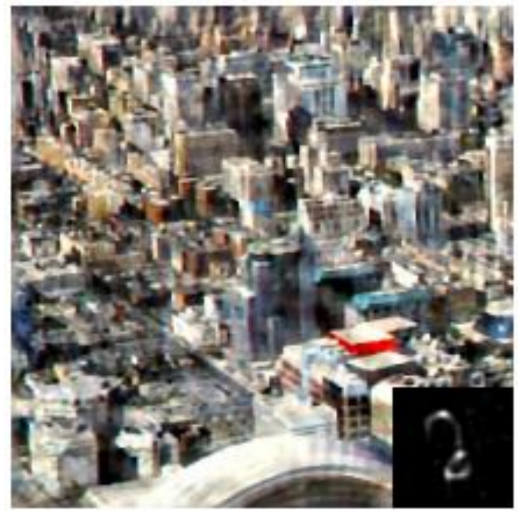
$$r(x) = \frac{\|\sum_{y \in N_{\mathrm{h}}(x)} \nabla \mathrm{B}(y)\|}{\sum_{y \in N_{\mathrm{h}}(x)} \|\nabla \mathrm{B}(y)\| + 0.5}$$

$$r(1) > r(2)$$



# Feature Extraction

Without elimination



With elimination

*Two-Phase Kernel Estimation for Robust Motion Deblurring. Li Xu et al.*

# SHIFT VARIANT DECONVOLUTION

# Geometric Model

- Camera movement
  - Rotation
  - Translation
- Rotation has significantly larger effect than translation

# Geometric Model

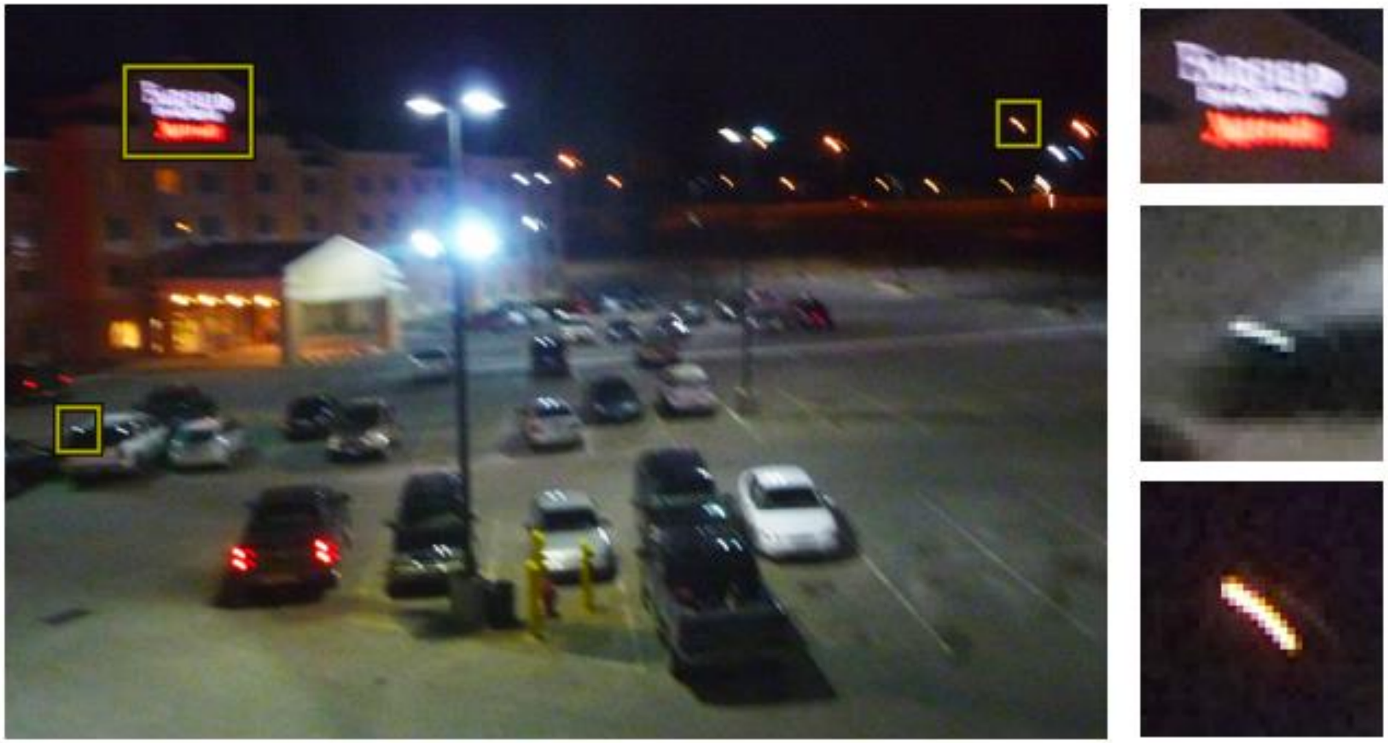


Figure 1. **Visible non-uniformity of blur in a shaken image.** Left: The blurry image. Right: Close-ups of different parts of the image. Note the differences in shape of the blur between the middle and bottom close-ups.

*Non-uniform Deblurring for Shaken images CVPR 2010, Oliver et al.*

# Block Blind Deconvolution

- initial from gyroscope
- Run blind deconvolution in each block

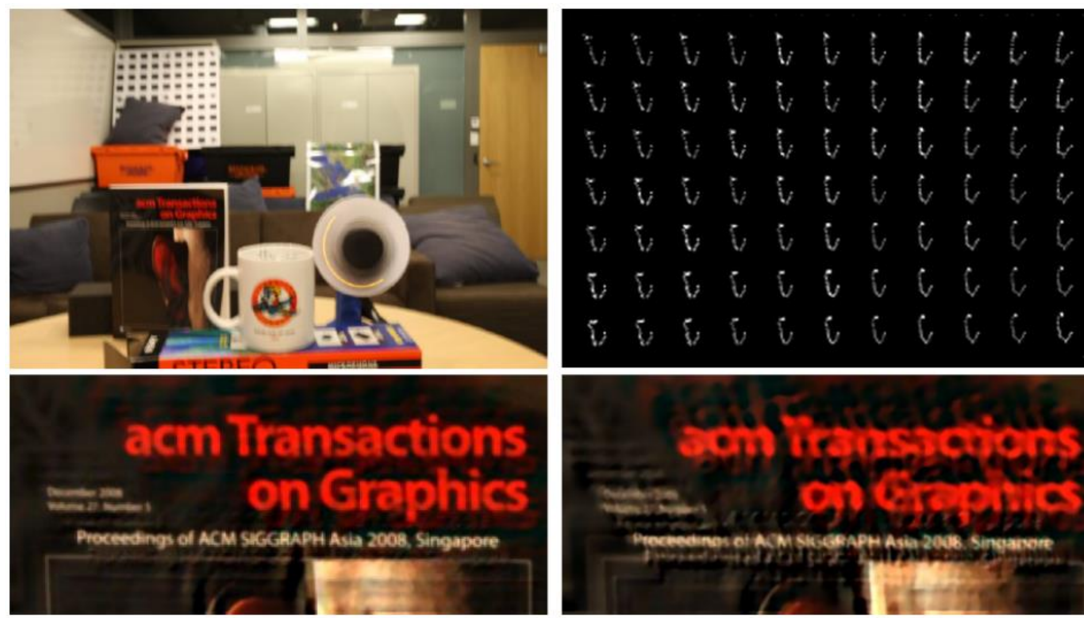



*image Deblurring using inertial Measurement Sensors SiGGRAPH 2010, Joshi et al.*

# Software

- Checkout deconvolution demo
  - http://sourceforge.net/projects/deconvdemo