Good Regions to Deblur

Zhe Hu and Ming-Hsuan Yang

Electrical Engineering and Computer Science
University of California at Merced
Merced, CA 95343

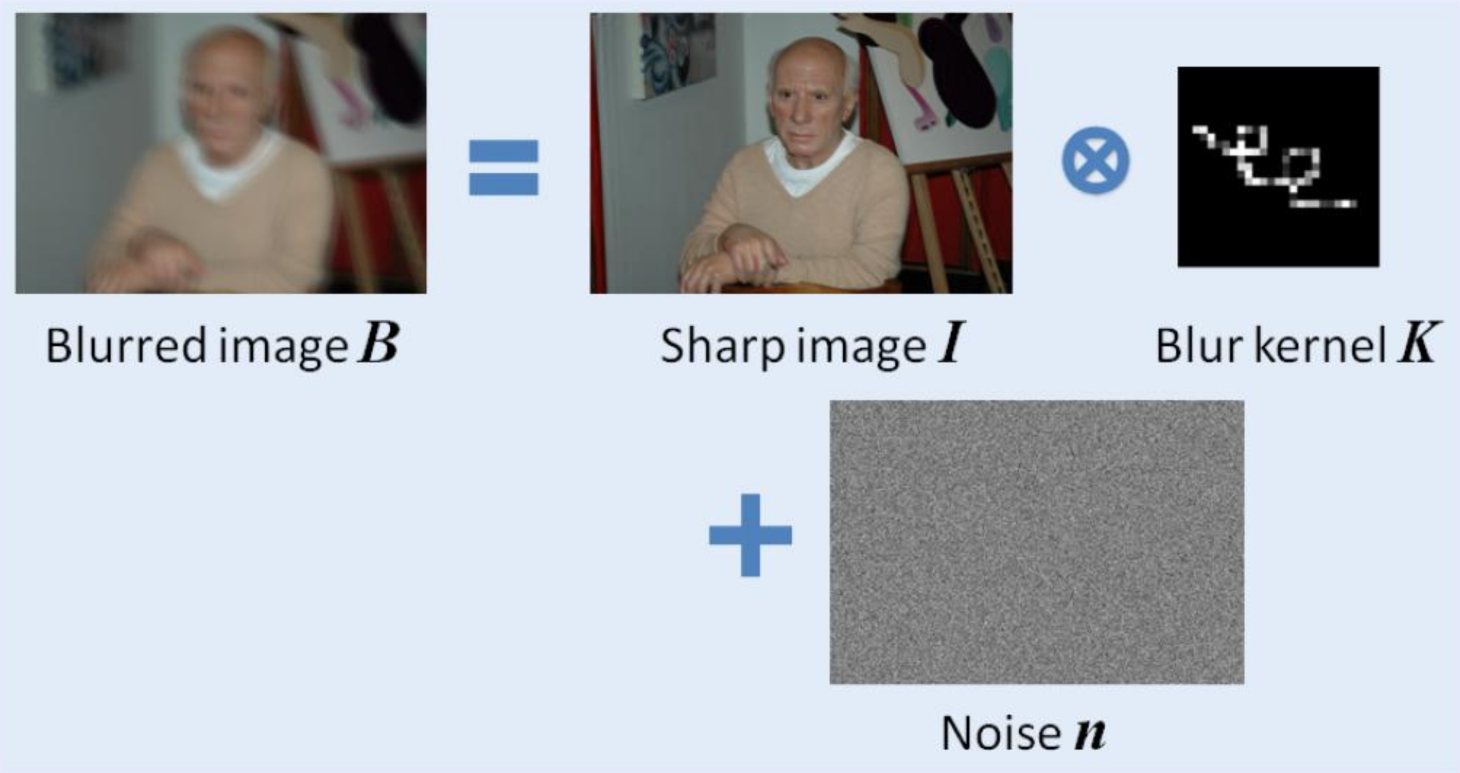
Image Blur

Caused by relative motion between the camera and the scene during the exposure time

Uniform Image Blur Model

Usually modeled by a linear convolution

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



[Shan et al. SIGGRAPH08]

Uniform Image Blur Model

- Non-blind deconvolution

- Blind deconvolution

[Shan et al. SIGGRAPH08]

- Image deblurring:
  - Kernel estimation
  - Non-blind deconvolution for image reconstruction

Related Work

Require prior knowledge and additional images or information

Prior Knowledge

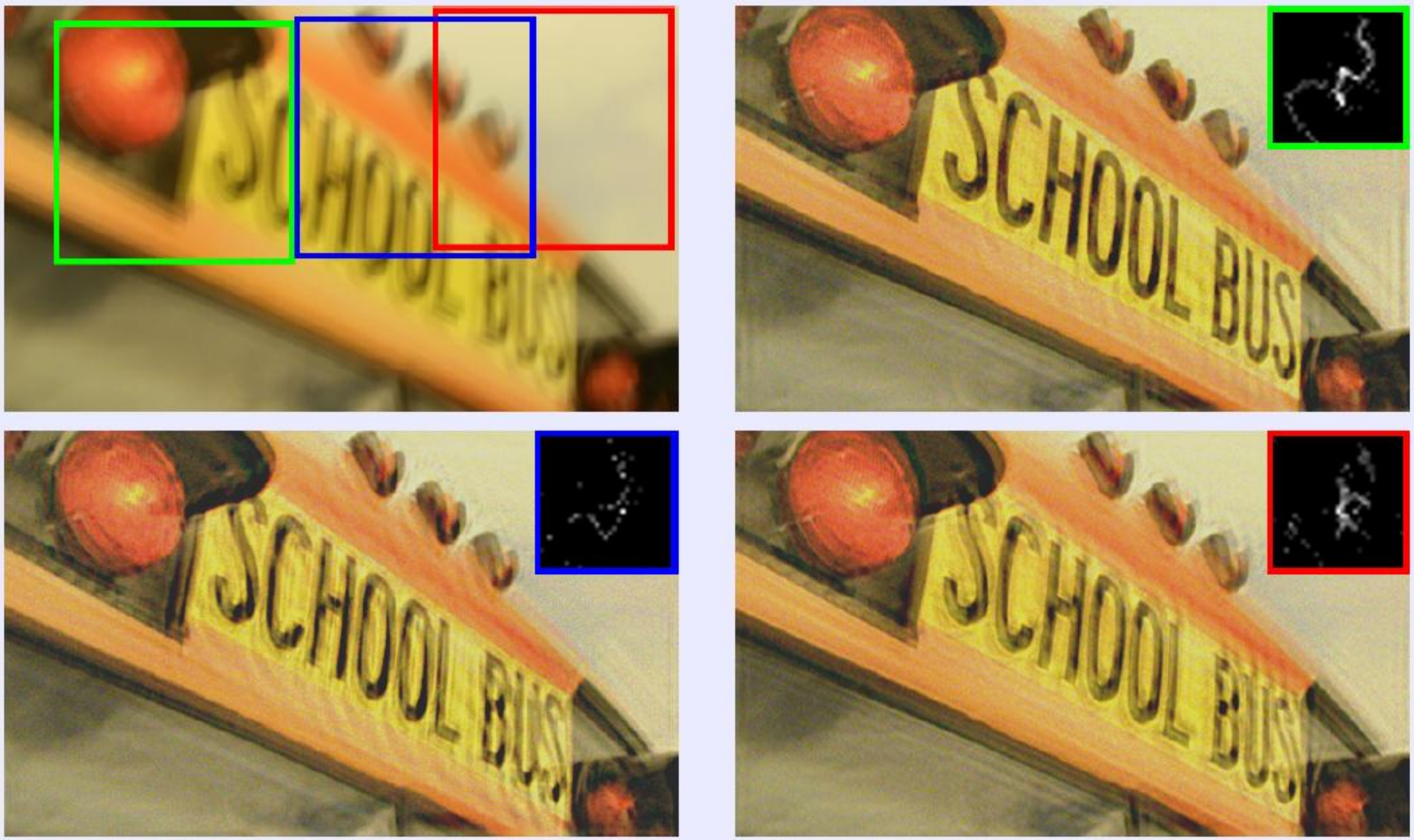
[Fergus et al. SIGGRAPH06], [Levin et al. NIPS06], [Jia CVPR07], [Shan et al. SIGGRAPH08], [Cai et al. CVPR09], [Cho & Lee SIGGRAPHAsia09], [Cho et al. CVPR10]

Additional Information

[Ben-Ezra et al. CVPR03], [Jia et al. ECCV04], [Yuan et al. CVPR07], [Raskar et al. SIGGRAPH06], [Levin et al. SIGGRAPH08]

Motivation

How to find good regions/subwindows for blur kernel estimation?



Motivation

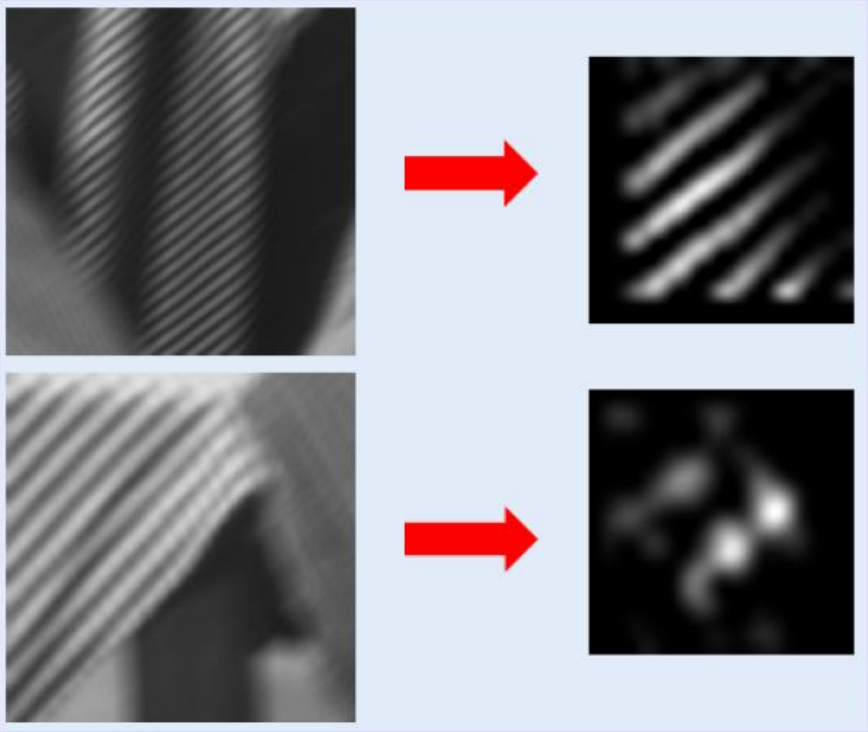
Why use regions to estimate kernel?

1. Computational efficiency for high-resolution image
2. Are all image regions useful?

Motivation

Why use regions to estimate kernel?

1. Computational efficiency for high-resolution image
2. Are all image regions useful?



Estimated kernels from cropped regions

Motivation

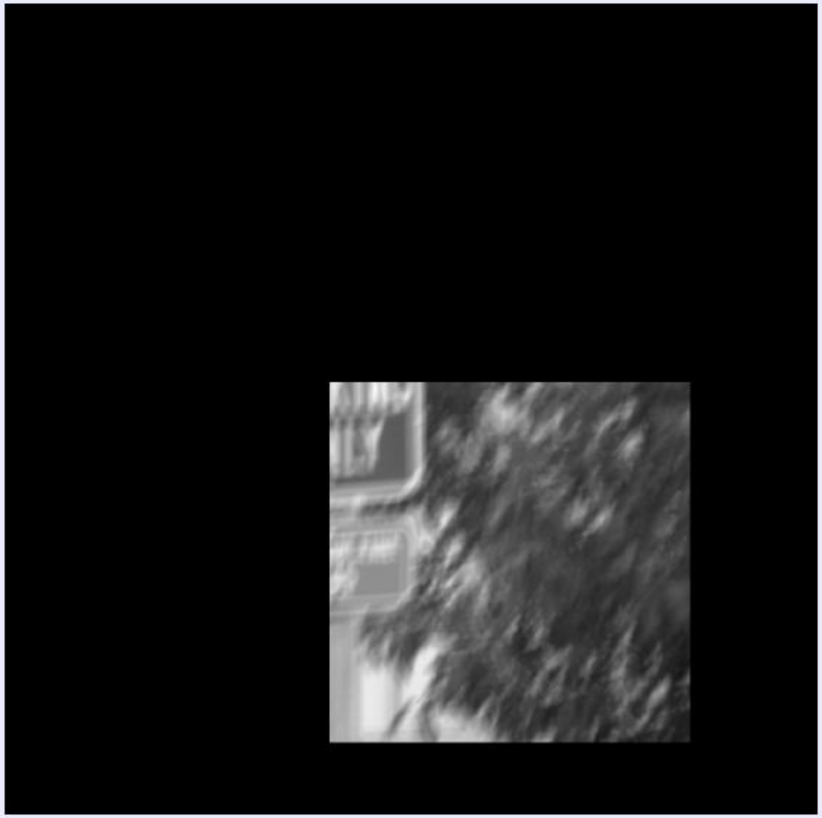
3. Using the whole image for better deblurring result?

Input blurry image and corresponding blur kernel

Estimated blur kernel using the whole image and deblurred image

Motivation

3. Using the whole image for better deblurring result?



Selected by the proposed method

Estimated kernel using the selected region and deblurred image

Motivation

What is good image structure for kernel estimation?

- Regions full with texture and edges [Fergus et al. SIGGRAPH06],[Levin et al. CVPR09]
- Salient edges with gradients of specific patterns [Joshi et al. CVPR08],[Cho et al. CVPR10],[Xu et al. ECCV10]

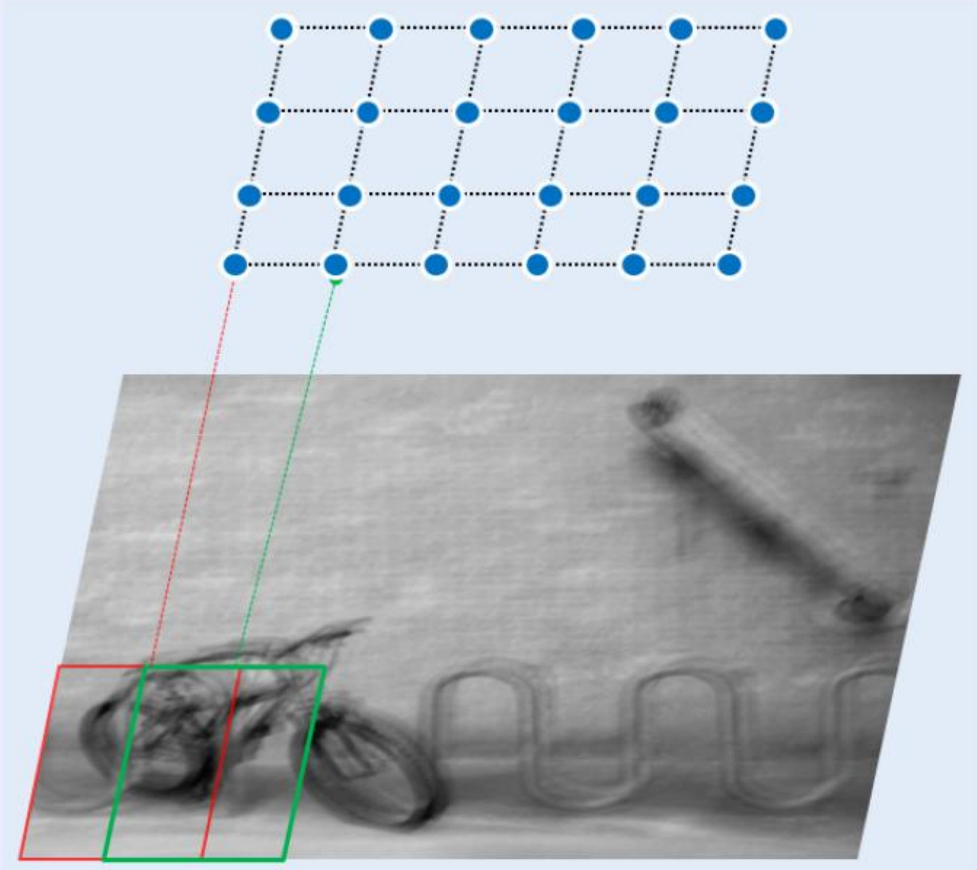
How to find good regions for kernel estimation?

- Whole image
- Manual selection of subwindow
- Regions with high intensity variance and low saturation [Fergus et al. SIGGRAPH06]
- Learn to find good regions (this work)

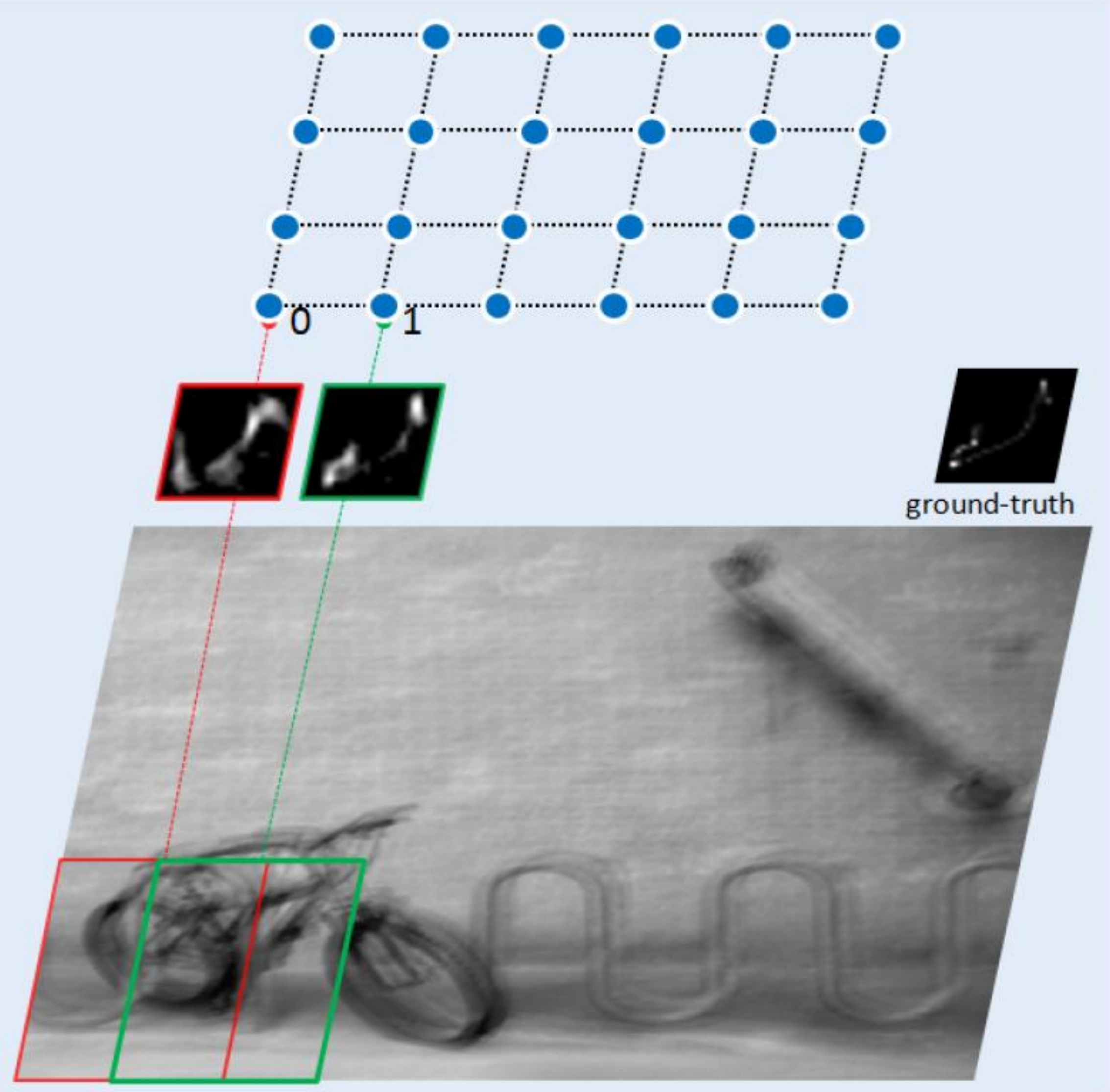
Learning Model

Which learning model to use?

- Closely overlapping subwindows share similar image structure
- Spatial correlation and local smoothness
- Conditional Random Field (CRF)



Learning Model



Learning Framework

Given labels $\mathbf{y}$ and observations $\mathbf{x}$, the conditional distribution $P(\mathbf{y}|\mathbf{x})$ can be written

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z}\exp\left(\sum_{i\in S} A_i(y_i,\mathbf{x}) + \sum_{i\in S}\sum_{j\in\mathcal{N}_i} I_{ij}(y_i,y_j,\mathbf{x})\right)$$

$A_i(y_i,\mathbf{x})$ and $I_{ij}(y_i,y_j,\mathbf{x})$: potential functions
S: node set in graph
$\mathcal{N}_i$: neighborhoods of node i
Z: partition function

Learning Framework

- Using log-likelihood and logistic functions similar to [Kumar and Hebert IJCV06]
- $\theta = \{\mathbf{w}, \mathbf{v}\}$ denotes the set of model parameters

Association Potential

$$A_i(y_i, \mathbf{x}) = \log P_1(y_i | h_i(\mathbf{x})))$$

$$P_1(y_i | h_i(\mathbf{x})) = \sigma(y_i \mathbf{w}^\top h_i(\mathbf{x}))$$

Interaction Potential

$$I(y_i, y_j, \mathbf{x}) = \log P_2(y_i y_j \mathbf{v}^\top \mu_{ij}(\mathbf{x}))$$

$$P_2(y_i, y_j | \mu_{ij}(\mathbf{x})) = \sigma(y_i y_j \mathbf{v}^\top \mu_{ij}(\mathbf{x}))$$

$$\mu_{ij}(\mathbf{x}) = |h_i(\mathbf{x}) - h_j(\mathbf{x})|$$

Feature Vector

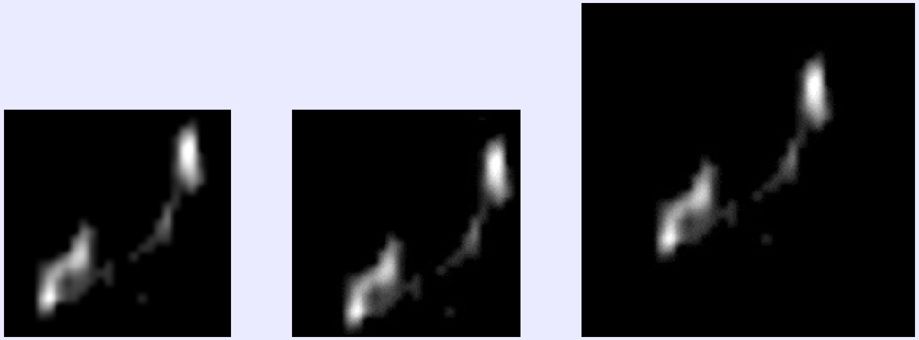
How to represent each region?

- Gabor filter response $f(\cdot)$
- Histogram of oriented gradients $g(\cdot)$
- Similar to [Xu et al. ECCV10], use mask $M(\cdot)$ to remove small edges
- Concatenate feature vectors $h(\mathbf{x}) = [f(\mathbf{x}), g(\mathbf{x}), f(M(\mathbf{x})), g(M(\mathbf{x}))]$

Kernel Similarity Metric

How to obtain training label?

- A good metric to compare estimated kernels with ground truth
- Shift and scale invariant



Reference Shifted Scaled

- Kernel similarity metric: maximum response of normalized cross-correlation between two kernels
- Obtain labels by thresholding kernel similarity values

Parameter Learning

- Maximum-likelihood to learn model parameter θ

$$\hat{\theta} = \arg\max_{\theta} \prod_{m} \prod_{i \in S_m} P(y_i^m | \mathbf{x}^m, \mathbf{y}_{\mathcal{N}_i}^m, \theta)$$

m: index of the training images

S_m: graph generated from the m-th image

- Conditional probability using pseudo-likelihood

$$P(y_i | \mathbf{x}, \mathbf{y}_{\mathcal{N}_i}, \theta) = \frac{1}{z_i} \exp(A_i(y_i, \mathbf{x}) + \sum_{j \in \mathcal{N}_i} I(y_i, y_j, \mathbf{x}))$$

with partition function

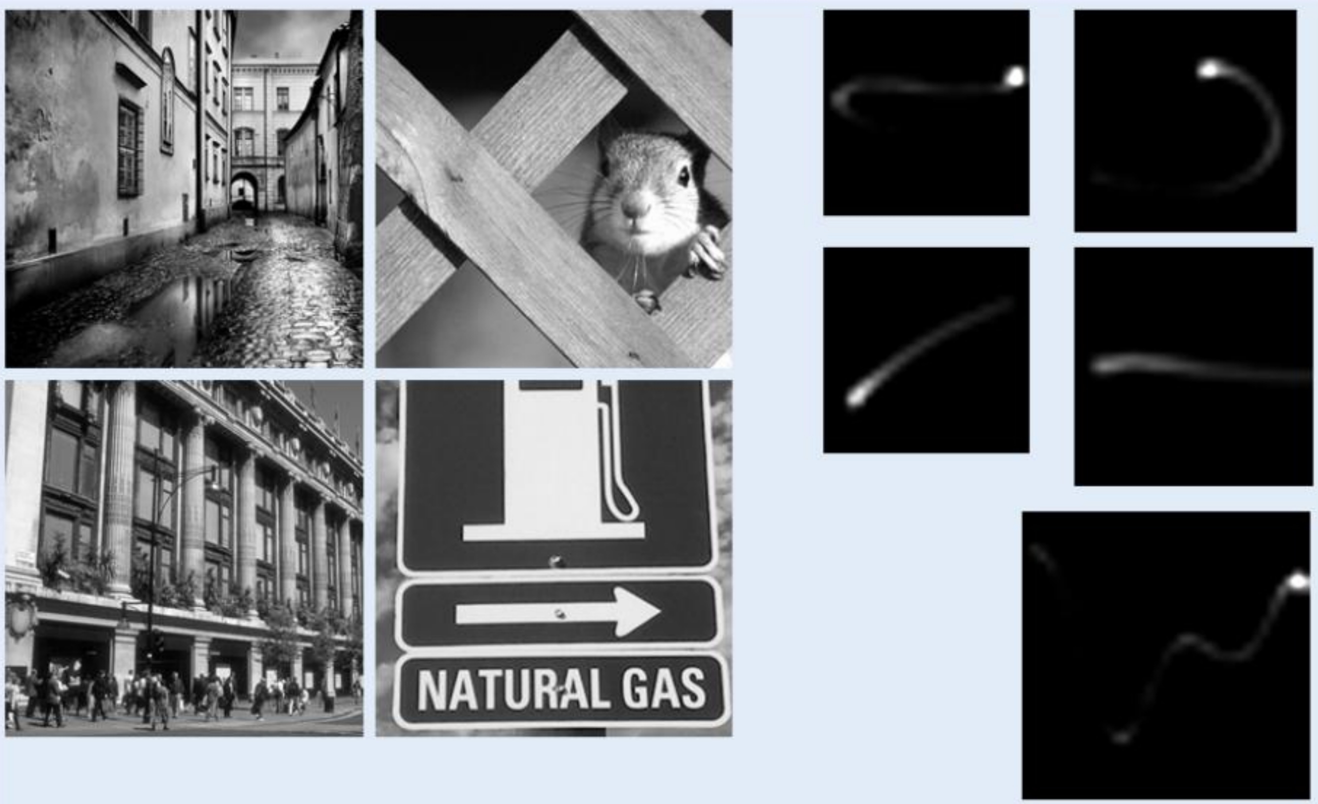
$z_i = \sum_{y_i \in \{-1,1\}} \exp(A_i(y_i, \mathbf{x}) + \sum_{j \in \mathcal{N}_i} I(y_i, y_j, \mathbf{x}))$

- Use BFGS to solve the optimization problem

Training Data

Obtain blurry images with ground-truth kernels as [Levin et al. CVPR09]

- 20 clear images $\times$ 20 blur kernels
- more than 484 nodes for each graph of a training image
- more than 484×400 regions analyzed

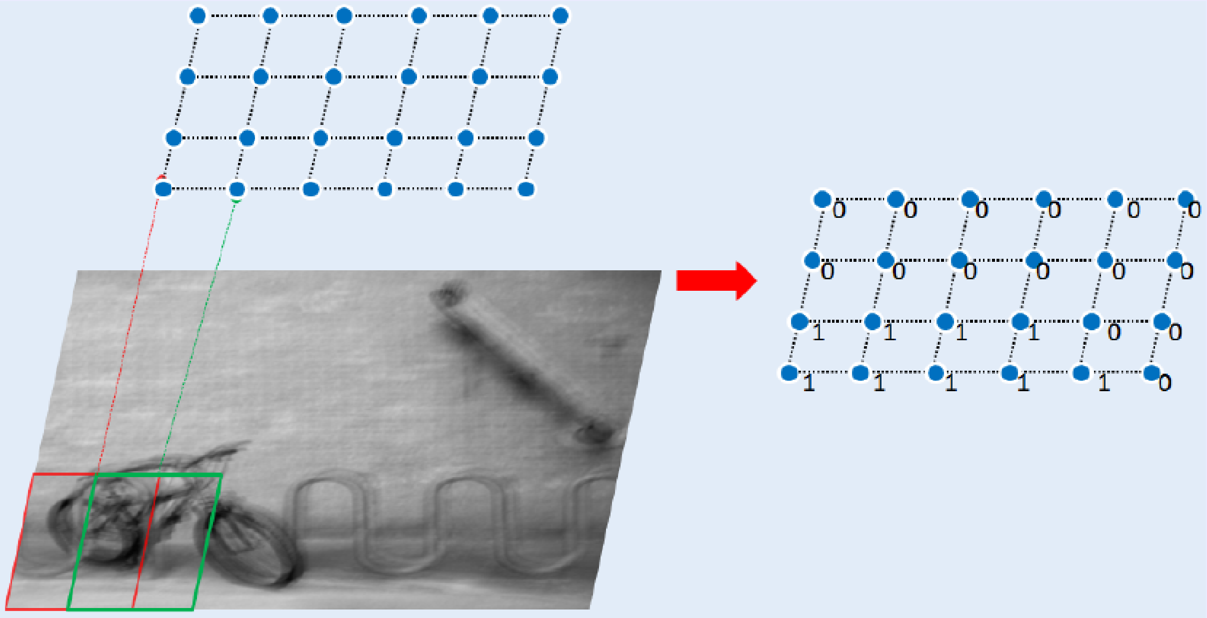


Inference

Scan each subwindow of a test image

- Use Loopy Belief Propagation (LBP) to infer good regions

$$\mathbf{y} \leftarrow \arg\max_{\mathbf{y}} P(\mathbf{y}|\mathbf{x})$$



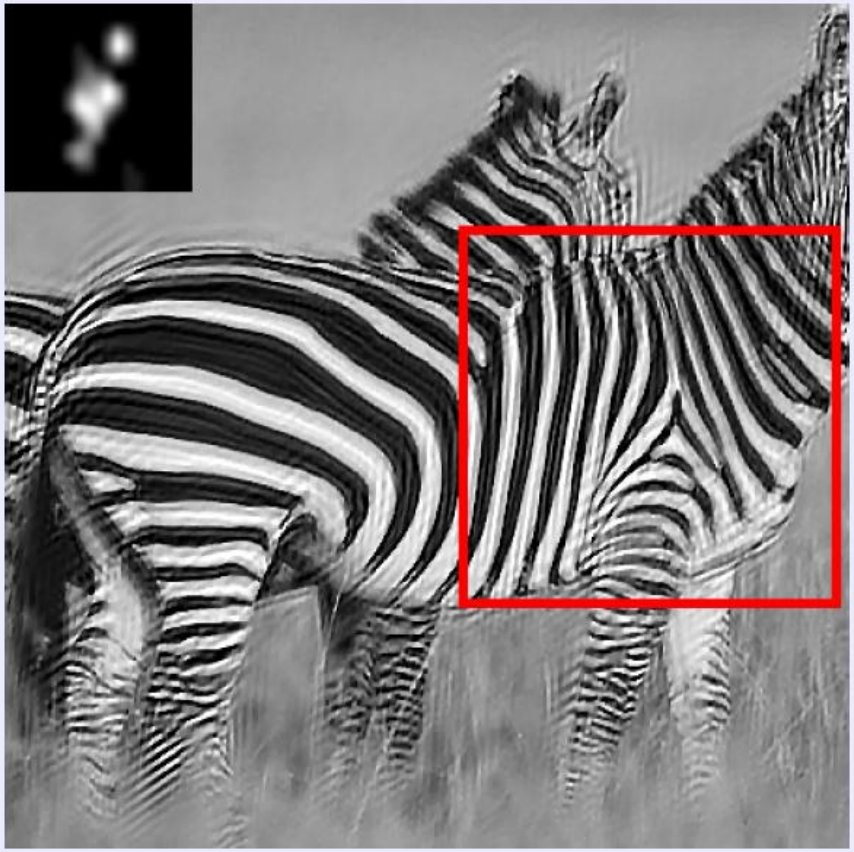
- Obtain labels with confidence values: top 1/top 10 subwindows as the region for kernel estimation

Experiments

- Use the learned CRF model to select good regions
- Estimate a blur kernel from each selected region (e.g., [Shan et al. SIGGRAPH08] and [Cho & Lee SIGGRAPHAsia09])
- Use the estimated kernel and [Shan et al. SIGGRAPH08] for non-blind deconvolution
- MATLAB implementation: 5 seconds to process an image of 450×450 pixels (3.40 GHz CPU and 16 GB RAM)
- More results and source code are available on `http://eng.ucmerced.edu/people/zhu/`

Experimental Results

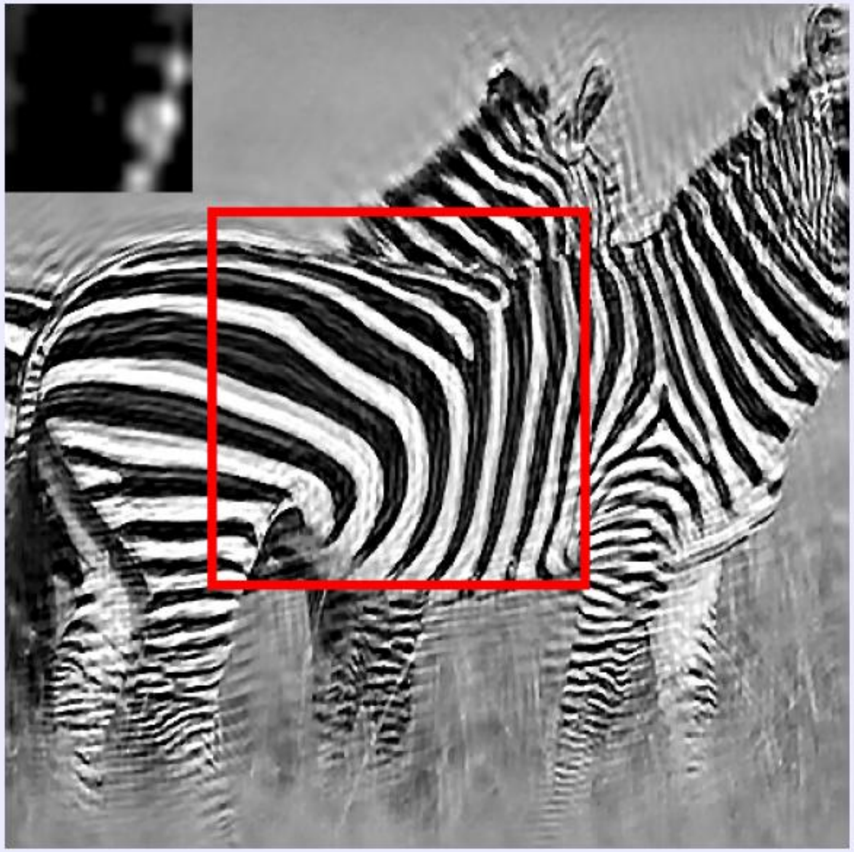
Input blurry image and corresponding ground truth blur kernel



Region: manual selection
Ker est: [Shan et al. SIGGRAPH08]

Experimental Results

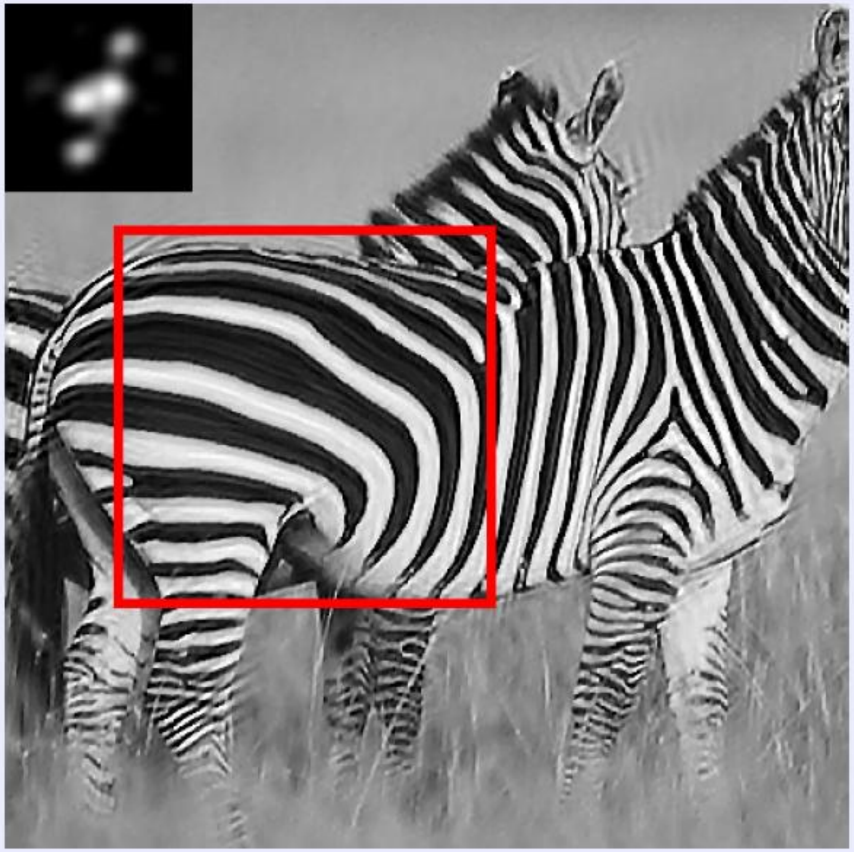
Input blurry image and corresponding ground truth blur kernel



Region: [Fergus et al. SIGGRAPH06]
Ker est: [Shan et al. SIGGRAPH08]

Experimental Results

Input blurry image and corresponding ground truth blur kernel



Region: proposed
Ker est: [Shan et al. SIGGRAPH08]

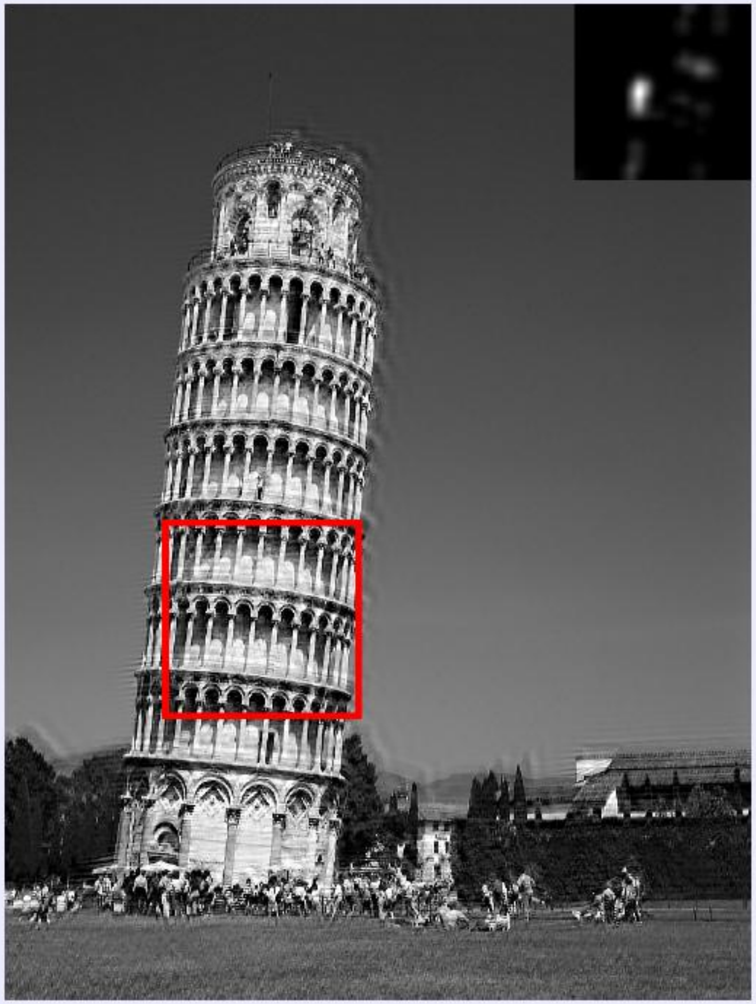
Experimental Results

Input blurry image and corresponding ground truth blur kernel

Region: whole image
Ker est: [Cho & Lee SIGGRAPHAsia09]

Experimental Results

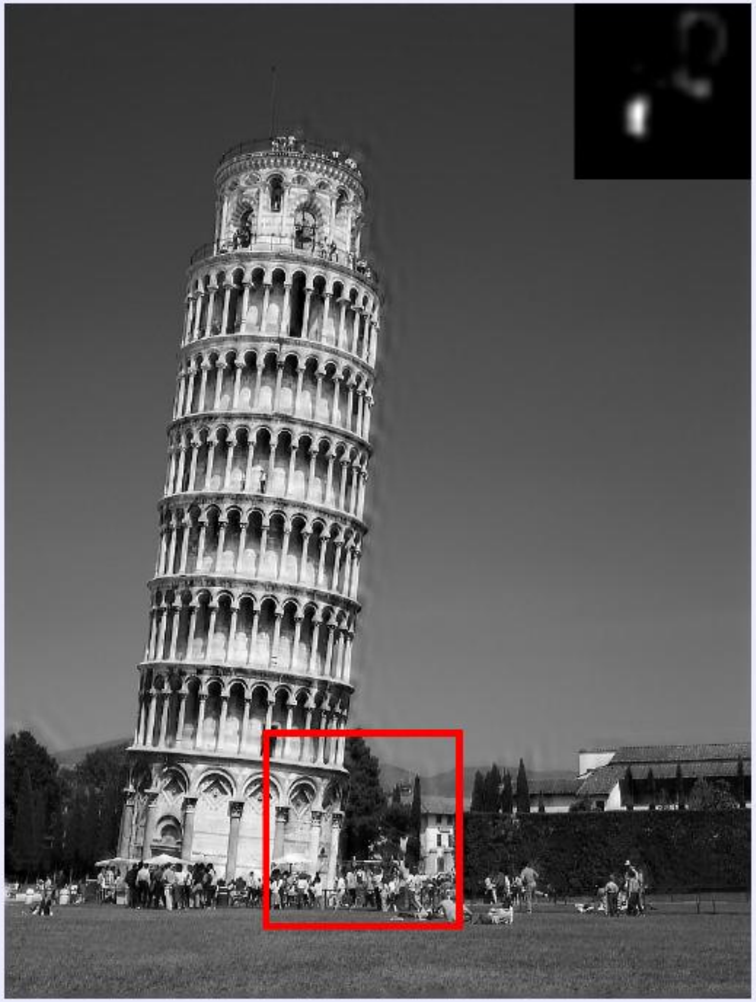
Region: whole image
Ker est: [Cho & Lee SIGGRAPHAsia09]



Region: [Fergus et al. SIGGRAPH06]
Ker est: [Cho & Lee SIGGRAPHAsia09]

Experimental Results

Region: whole image
Ker est: [Cho & Lee SIGGRAPHAsia09]



Region: proposed
Ker est: [Cho & Lee SIGGRAPHAsia09]

Results Using Top Ten Subwindows

Input blurry image and corresponding ground truth blur kernel

Region: whole image
Ker est: [Cho & Lee SIGGRAPHAsia09]

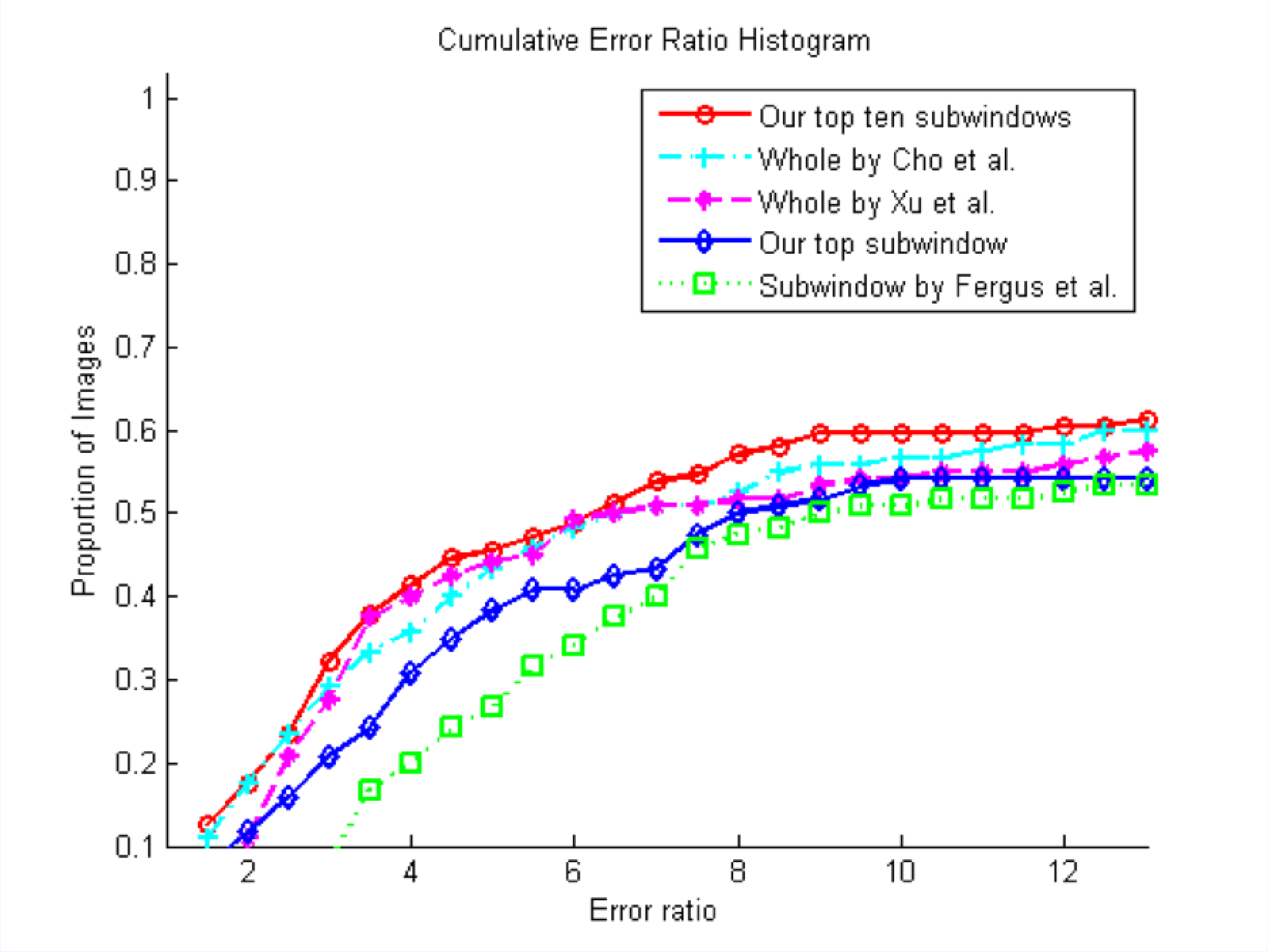
Results Using Top Ten Subwindows

Inferred region using the top ten subwindows

Region: top ten subwindows
Ker est: [Cho & Lee SIGGRAPHAsia09]

Quantitative Comparison

- 10 clear images $\times$ 12 blur kernels for test
- Error ratio: $||I_r - I_g||^2 / ||I_{k_g} - I_g||^2$ [Levin et al. CVPR09]



Concluding Remarks

Learning good regions to deblur:

- Exploiting informative image structure for deblurring
- Selecting good regions for effective deblurring
- Can be used for non-uniform deblurring