

Fast Motion Deblurring

Supplementary Material I

Sunghyun Cho Seungyong Lee

POSTECH

Algorithm – Fast motion deblurring

procedure FastDeblur(B)

$L \leftarrow B$

for coarse to fine **do**

 resize K , L , and B according to the current scale

for iter $\leftarrow 1$, max_iter **do**

$\{P_x, P_y\} \leftarrow \text{predict}(L)$

$K \leftarrow \text{estimate_kernel}(\{P_x, P_y\}, B)$

$L \leftarrow \text{deconvolve}(B, K)$

if K is not updated **then**

break

end if

end for

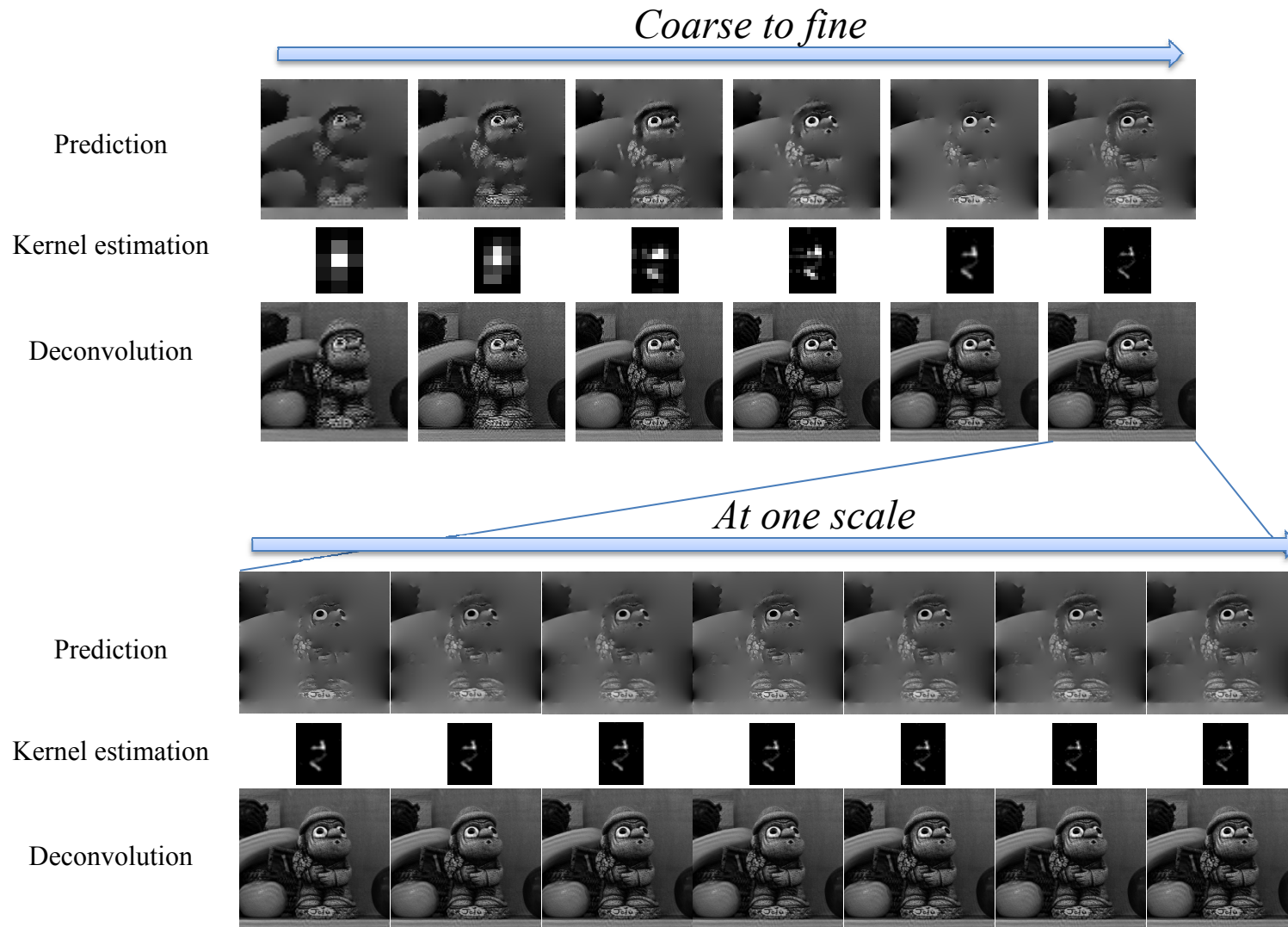
end for

$L \leftarrow \text{deconvolve_final}(B, K)$

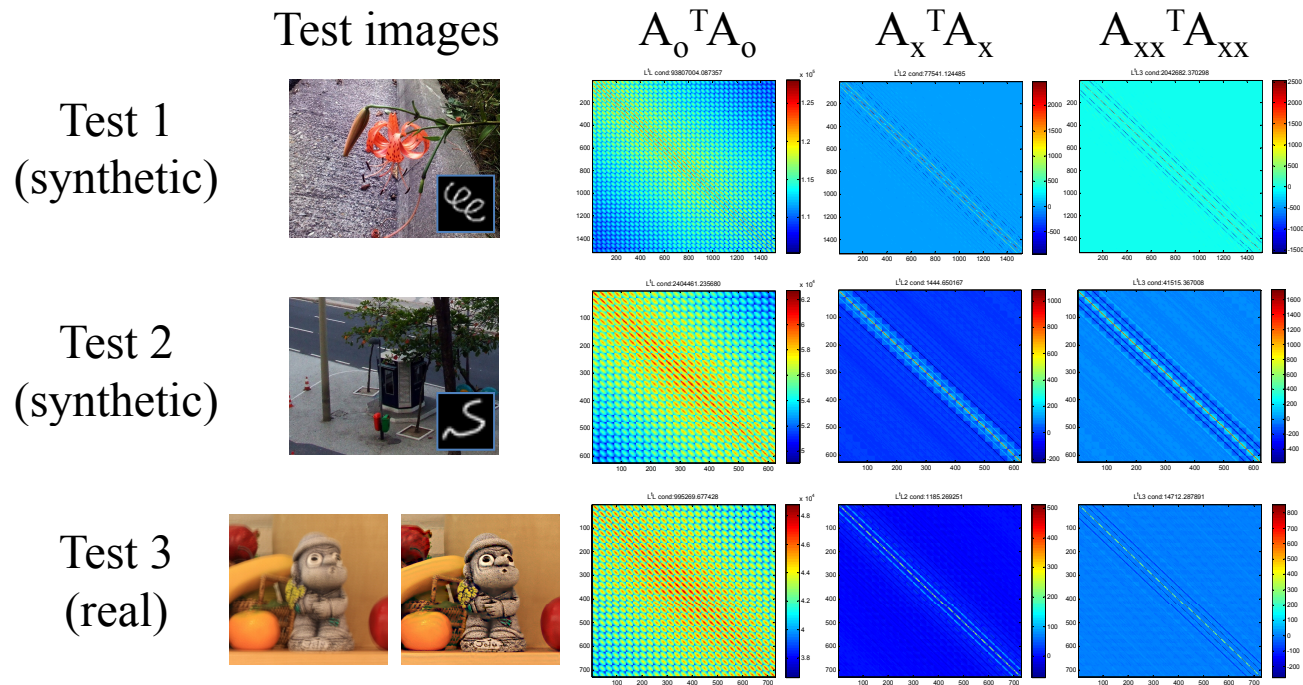
return L

end procedure

Intermediate results of a real example



More experiments on the condition number of $A^T A$



Condition numbers	$A_0^T A_0$	$A_x^T A_x$	$A_{xx}^T A_{xx}$
Test 1	9.3807×10^7	7.7541×10^4	2.0427×10^6
Test 2	2.4045×10^6	1.4447×10^3	4.1515×10^4
Test 3	9.9527×10^5	1.1853×10^3	1.4712×10^4

Synthetic examples (1)

Image size	1024 x 1024
Kernel size	25 x 25
Kernel estimation	2.062 sec
Final deconvolution	2.079 sec
Total processing time	4.156 sec



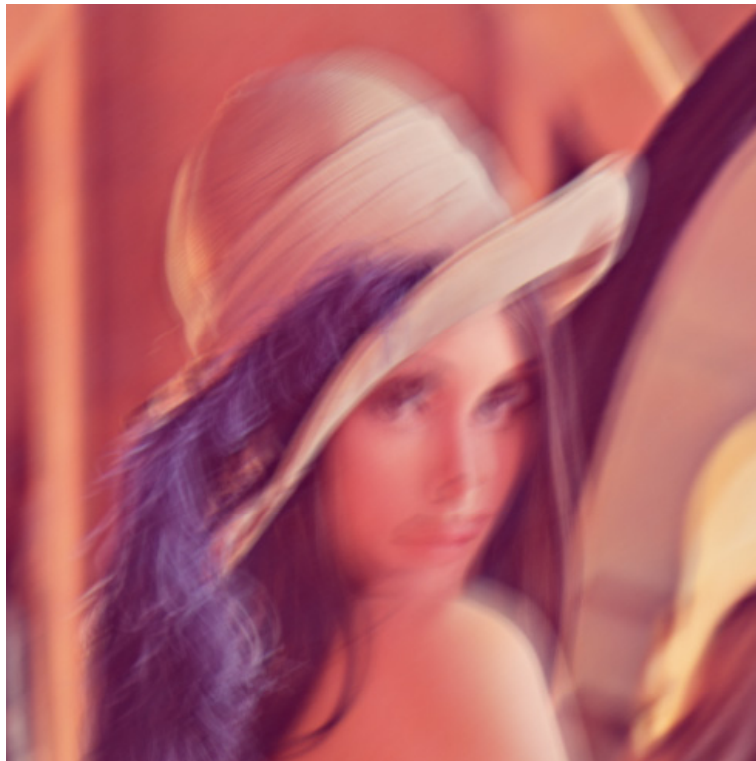
Blurry input



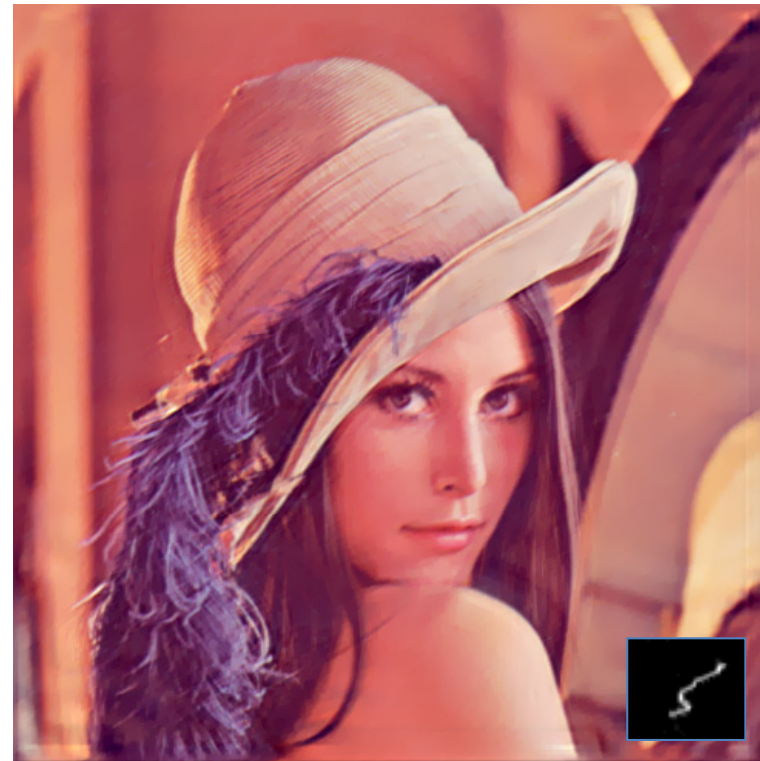
Our result

Synthetic examples (2)

Image size	491 x 491
Kernel size	43 x 37
Kernel estimation	0.250 sec
Final deconvolution	0.141 sec
Total processing time	0.391 sec



Blurry input



Our result

Synthetic examples (3)

Image size	700 x 525
Kernel size	45 x 33
Kernel estimation	0.375 sec
Final deconvolution	1.187 sec
Total processing time	1.562 sec



Blurry input



Our result

Comparison with previous methods (1)



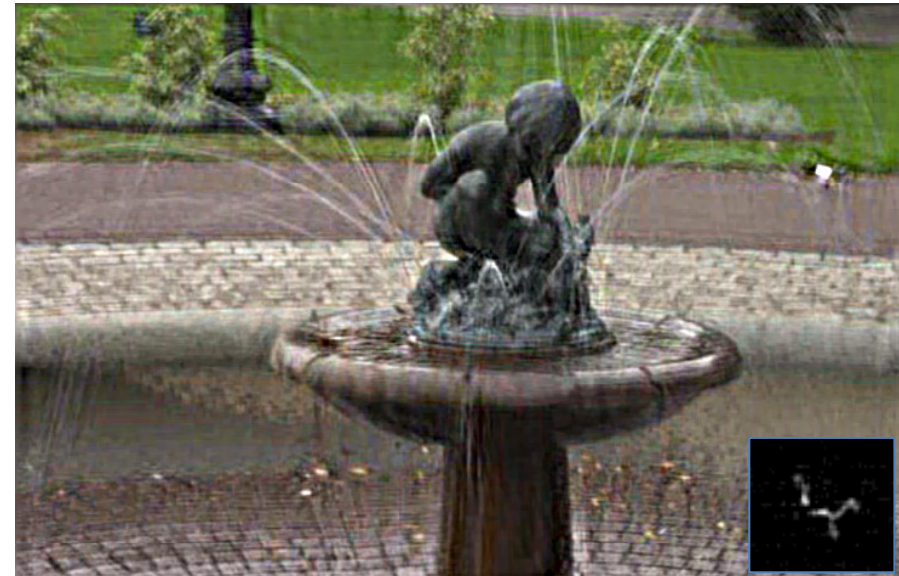
Blurry input

Image size	732 x 474
Kernel size	53 x 49
Kernel estimation	0.359 sec
Final deconvolution	1.375 sec
Total processing time	1.734 sec

- Even though there is a strongly saturated area in the upper right part of the image, our method can produce a deblurring result without specifying a region for kernel estimation. In [Fergus et al. 2006], for correct deblurring, a user needs to specify a region with no saturation.



[Fergus et al. 2006]



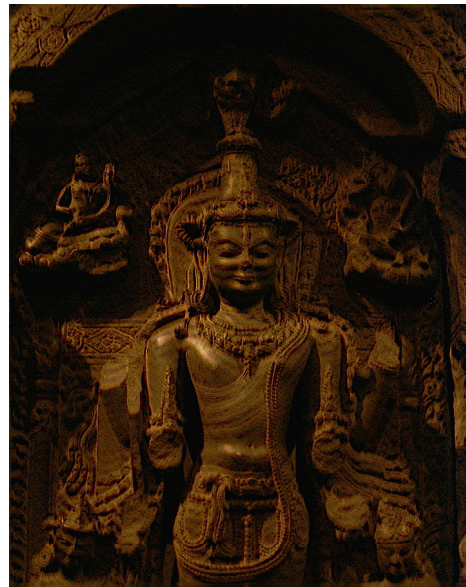
Our method

Comparison with previous methods (2)

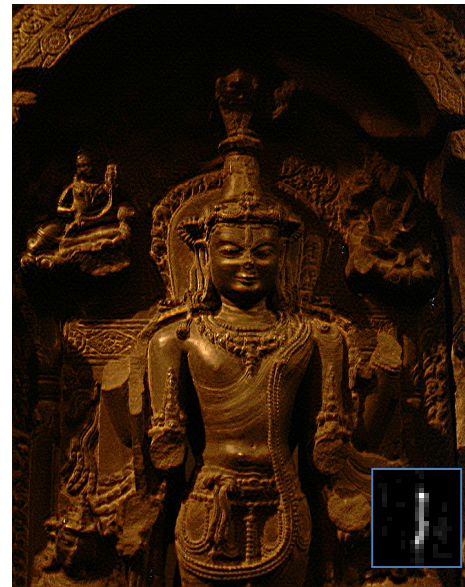


Blurry input

Image size	724 x 905
Kernel size	12 x 21
Kernel estimation	0.484 sec
Final deconvolution	1.291 sec
Total processing time	1.775 sec



[Fergus et al. 2006]



Our method

Comparison with previous methods (3)



Blurry input (Picasso)

Image size	800 x 532
Kernel size	27 x 19
Kernel estimation	0.375 sec
Final deconvolution	0.204 sec
Total processing time	0.594 sec



[Shan et al. 2008]



Our method

Comparison with previous methods (4)



Blurry input (statue)

Image size	903 x 910
Kernel size	25 x 25
Kernel estimation	0.594 sec
Final deconvolution	0.359 sec
Total processing time	0.953 sec

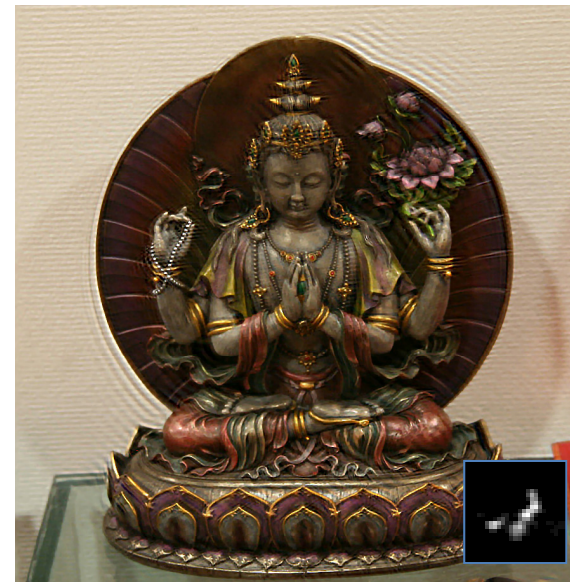
- Results in this slide have been obtained without additional ringing suppression described in the supplementary material provided by the authors of [Shan et al. 2008].

(http://www.cse.cuhk.edu.hk/~leojia/projects/motion_deblurring/implementation.pdf)

- Please refer to the paper if you want to see the results with additional ringing suppression.



[Shan et al. 2008]



Our method

Comparison with previous methods (5)

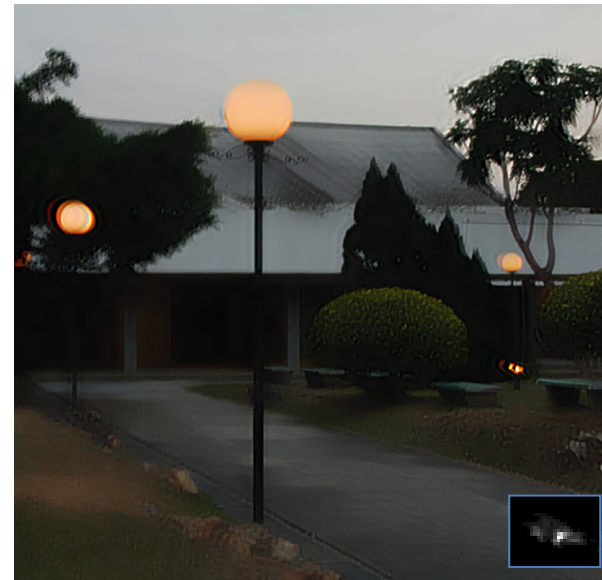


Blurry input (night)

Image size	836 x 804
Kernel size	27 x 21
Kernel estimation	0.500 sec
Final deconvolution	0.969 sec
Total processing time	1.469 sec

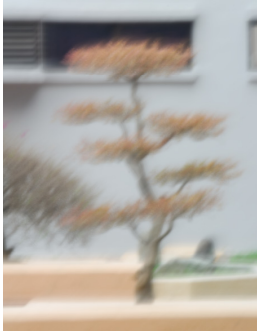


[Shan et al. 2008]



Our method

Comparison with previous methods (6)



Blurry input (red tree)

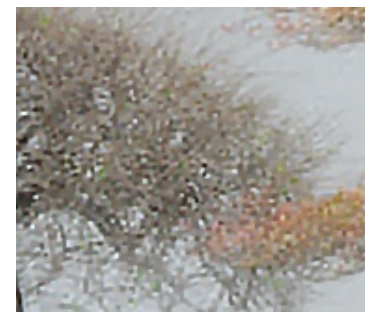
Image size	454 x 588
Kernel size	27 x 27
Kernel estimation	0.250 sec
Final deconvolution	0.140 sec
Total processing time	0.406 sec



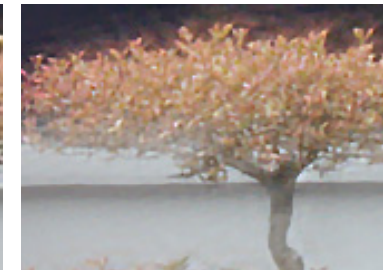
[Shan et al. 2008]



Our method



[Shan et al. 2008]



Our method

Comparison with previous methods (7)



Blurry input

Image size	508 x 500
Kernel size	49 x 35
Kernel estimation	0.297 sec
Final deconvolution	0.875 sec
Total processing time	1.172 sec



[Shan et al. 2008]



Our method

Comparison with previous methods (8)



Blurry input

Image size	454 x 588
Kernel size	53 x 89
Kernel estimation	0.297 sec
Final deconvolution	1.766 sec
Total processing time	2.063 sec



[Shan et al. 2008]



Our method

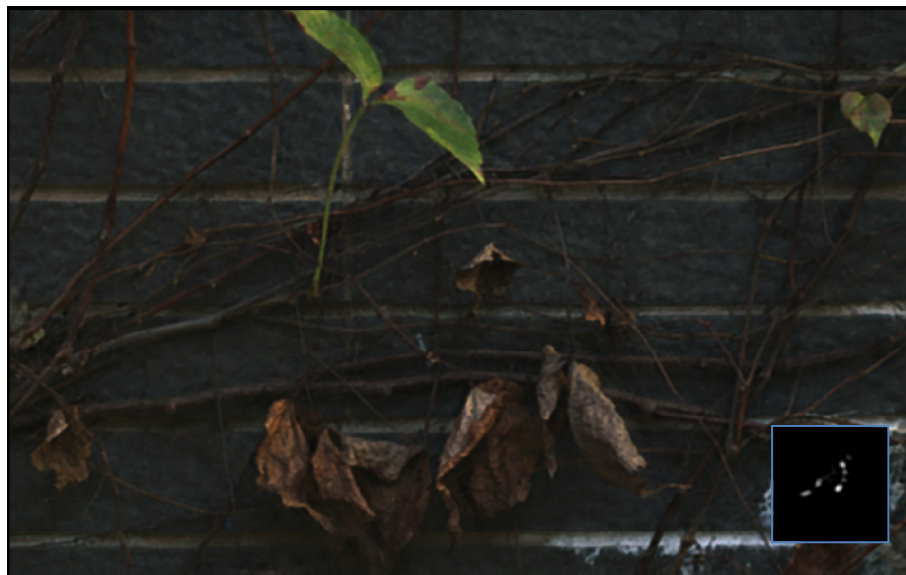
Comparison with previous methods (9)



Blurry input

Image size	1011 x 635
Kernel size	99 x 99
Kernel estimation	1.188 sec
Final deconvolution	11.016 sec
Total processing time	12.204 sec

- Note that two blurred images are needed in [Chen et al. 2008], while our method can deblur a single input image.



[Chen et al. 2008]



Our method

Real examples (1)

Image size	900 x 741
Kernel size	21 x 23
Kernel estimation	0.516 sec
Final deconvolution	1.360 sec
Total processing time	1.876 sec



Blurry input



Our result

Real examples (2)

Image size	918 x 1009
Kernel size	21 x 23
Kernel estimation	0.953 sec
Final deconvolution	2.829 sec
Total processing time	3.782 sec



Blurry input



Our result

Real examples (3)

Image size	800 x 766
Kernel size	45 x 45
Kernel estimation	0.516 sec
Final deconvolution	2.015 sec
Total processing time	2.531 sec



Blurry input



Our result

Real examples (4)

Image size	1329 x 955
Kernel size	43 x 53
Kernel estimation	1.297 sec
Final deconvolution	0.891 sec
Total processing time	2.204 sec



Blurry input



Our result

Limitations

- Saturation

- Since saturation breaks the linearity of the convolution-based blur model, correct kernel estimation is difficult.
- Even if a correct kernel is estimated, deconvolution of saturated pixels is a difficult problem.



Blurry input

Our results

* We thank Pusan National University for providing the input blurry image of the example in the first row.

Limitations

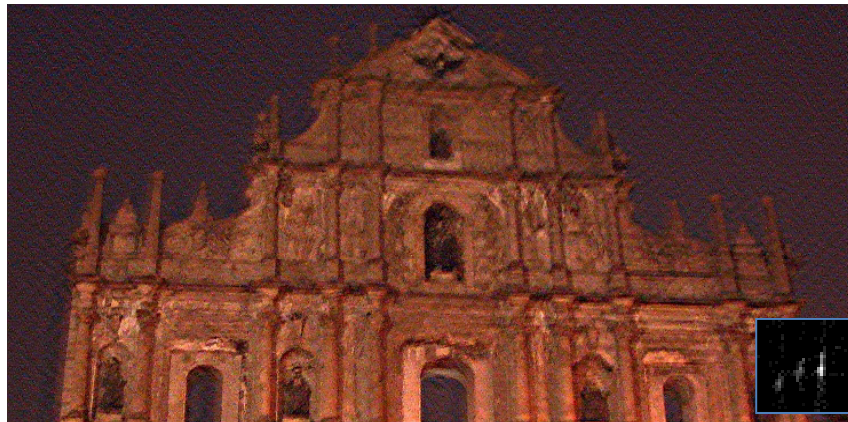
- Severe noise

- Since severe noise breaks the linearity of the convolution-based blur model, correct kernel estimation is difficult.
- Even if a correct kernel is estimated, deconvolution of a severely noisy image is a difficult problem.

Blurry input



Our result



Limitations

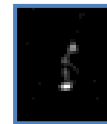
- Spatially varying blur
 - Caused by depth differences, moving objects, and rotational motions of a camera.

Rotational motion of a camera



Kernel estimated from the **left** part of the input image and the deblurring result using the estimated kernel.

Kernel estimated from the **right** part of the input image and the deblurring result using the estimated kernel.



Two estimated kernels are different.



Thank You.

<http://cg.postech.ac.kr>