

From local to global: Edge profiles to camera motion in blurred images

SUPPLEMENTARY MATERIAL

In the supplemental material, we begin with a high level description of the contents of the main paper. We then provide complete derivation of the relation between edge profiles and MDF (Section 3) as well as proofs for claims 1, 2, and 3 in the main paper. This is followed by additional implementation details, performance comparisons for large blur kernels, and run-time comparisons for all the methods. We also provide performance comparisons on additional example images taken from existing real image datasets.

S1. High level description of the contents in main paper

In Section 3, we derive the relation between the local edge profiles present in a motion blurred image and the underlying global camera motion. However, there are practical difficulties in using such a relation to perform camera motion estimation. To solve issues related to alignment of edge-profiles, we propose a solution in Section 4.1, and theoretically validate the applicability of the solution under the assumption of small in-plane rotations. In Section 4.2, a computationally efficient form of an edge-profile constraint is derived for direct camera motion estimation. In Section 5, we embed this constraint into an existing blind deblurring method to improve its performance further.

S2. Additional derivations and proofs

S2.1. Edge profiles to camera motion

We have already discussed in the main paper that, the PSF at any location in the blurred image can be related to the MDF as

$$k(\mathbf{x}, \mathbf{u}) = \int \Psi(\gamma) \delta(\mathbf{u} - (H^\gamma \mathbf{x} - \mathbf{x})) d\gamma \quad (\text{S1})$$

Also, the differential of edge profile (along the direction perpendicular to θ i.e., $\theta + \pi/2$) at \mathbf{x} can be expressed as the Radon transform of the PSF at \mathbf{x} along θ as

$$dE_{\theta, \mathbf{x}}(\rho) = \int \int k_\mathbf{x}(u, v) \delta(\rho - u \cos \theta - v \sin \theta) du dv \quad (\text{S2})$$

Combining the relations in Eq. (S1) and Eq. (S2) we can obtain the relation between MDF and the edge profile as

$$dE_{\theta, \mathbf{x}}(\rho) = \int \int \left(\int_{\gamma \in \Gamma} \Psi(\gamma) \delta(\mathbf{u} - (H^\gamma \mathbf{x} - \mathbf{x})) d\gamma \right) \delta(\rho - u \cos \theta - v \sin \theta) du dv \quad (\text{S3})$$

$$= \int_{\gamma \in \Gamma} \Psi(\gamma) \left(\int \int \delta(\mathbf{u} - (H^\gamma \mathbf{x} - \mathbf{x})) \delta(\rho - u \cos \theta - v \sin \theta) du dv \right) d\gamma \quad (\text{S4})$$

$$= \int_{\gamma \in \Gamma} \Psi(\gamma) \delta(\rho - (H_x^\gamma \mathbf{x} - x) \cos \theta - (H_y^\gamma \mathbf{x} - y) \sin \theta) d\gamma \quad (\text{S5})$$

S2.2. Proof for claim 1

Claim 1: For model_1 , if the in-plane rotational motion undergone by the camera is small then it can be shown that the centroid of MDF vector will correspond to the centroid of PSF i.e., the centroid pose of the camera obtained from MDF will map points to the centroid of PSFs generated by the camera motion.

Proof: If N_T is the total number of poses over which the MDF is defined, then the centroid $(t_x^c, t_y^c, \theta_z^c)$ of MDF vector w can be found as follows

$$t_x^c = \sum_{p=1}^{N_T} t_x^p w(t_x^p, t_y^p, \theta_z^p) \quad (\text{S6})$$

$$t_y^c = \sum_{p=1}^{N_T} t_y^p w(t_x^p, t_y^p, \theta_z^p) \quad (\text{S7})$$

$$\theta_z^c = \sum_{p=1}^{N_T} \theta_z^p w(t_x^p, t_y^p, \theta_z^p) \quad (\text{S8})$$

The relation between the coordinates of both latent image and its warped form (warped according to the camera pose $(t_x^p, t_y^p, \theta_z^p)$) is given by

$$\begin{pmatrix} x^p \\ y^p \\ 1 \end{pmatrix} = H^p \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (\text{S9})$$

$$= K \begin{pmatrix} \cos(\theta_z^p) & \sin(\theta_z^p) & \frac{t_x^p}{d} \\ -\sin(\theta_z^p) & \cos(\theta_z^p) & \frac{t_y^p}{d} \\ 0 & 0 & 1 \end{pmatrix} K^{-1} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (\text{S10})$$

$$= \begin{pmatrix} \cos(\theta_z^p) & \sin(\theta_z^p) & \frac{qt_x^p}{d} \\ -\sin(\theta_z^p) & \cos(\theta_z^p) & \frac{qt_y^p}{d} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (\text{S11})$$

where d is the perpendicular distance of the scene-plane from the image-plane. K is the intrinsic matrix of the camera defined by the focal length q .

$$K = \begin{pmatrix} q & 0 & 0 \\ 0 & q & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (\text{S12})$$

Let us denote the MDF weight $w(t_x^p, t_y^p, \theta_z^p)$ corresponding to the pose p as w^p for simplicity. Note that w^p is a scalar value representing the fraction of time that the camera spent at pose p . The x -coordinate of the centroid (u_c, v_c) of psf $k_{\mathbf{x}}$ is given by,

$$u_c = \sum_u \sum_v u k_{\mathbf{x}}(u, v) \quad (\text{S13})$$

Consider the discrete form of Eq. (S1) given by

$$k_{\mathbf{x}}(u, v) = k(\mathbf{x}, \mathbf{u}) = \sum_{p=1}^{N_T} w^p \delta(\mathbf{u} - (H^p \mathbf{x} - \mathbf{x})) \quad (\text{S14})$$

From Eq. (S13) and Eq. (S14) we can perform the following simplifications.

$$u_c = \sum_u \sum_v u \left(\sum_{p=1}^{N_T} w^p \delta(\mathbf{u} - (H^p \mathbf{x} - \mathbf{x})) \right) \quad (\text{S15})$$

$$= \sum_{p=1}^{N_T} w^p \left(\sum_u \sum_v u \delta(\mathbf{u} - (H^p \mathbf{x} - \mathbf{x})) \right) \quad (\text{S16})$$

$$= \sum_{p=1}^{N_T} w^p (H_x^p \mathbf{x} - x) \quad (\text{S17})$$

$$= \sum_{p=1}^{N_T} w^p H_x^p \mathbf{x} - \sum_{p=1}^{N_T} w^p x \quad (\text{S18})$$

$$= \sum_{p=1}^{N_T} w^p H_x^p \mathbf{x} - x \quad (\text{S19})$$

where to arrive at Eq. (S19) we have used the fact that MDF vector (w) sums to 1. When the in-plane rotational motion of camera is small, we use the approximations $\sin(\theta) \simeq \theta$ and $\cos(\theta) \simeq 1$. Now using Eq. (S11) we can obtain the following results.

$$\sum_{p=1}^{N_T} w^p H_x^p \mathbf{x} = \left(\sum_{p=1}^{N_T} H^p \cdot w^p \right)_x \mathbf{x} \quad (\text{S20})$$

$$= \left(\sum_{p=1}^{N_T} \begin{pmatrix} \cos(\theta_z^p) & \sin(\theta_z^p) & \frac{qt_x^p}{d} \\ -\sin(\theta_z^p) & \cos(\theta_z^p) & \frac{qt_y^p}{d} \\ 0 & 0 & 1 \end{pmatrix} w^p \right)_x \mathbf{x} \quad (\text{S21})$$

$$= \begin{pmatrix} \sum_{p=1}^{N_T} \cos(\theta_z^p) w^p & \sum_{p=1}^{N_T} \sin(\theta_z^p) w^p & \frac{q \sum_{p=1}^{N_T} t_x^p w^p}{d} \\ -\sum_{p=1}^{N_T} \sin(\theta_z^p) w^p & \sum_{p=1}^{N_T} \cos(\theta_z^p) w^p & \frac{q \sum_{p=1}^{N_T} t_y^p w^p}{d} \\ 0 & 0 & 1 \end{pmatrix}_x \mathbf{x} \quad (\text{S22})$$

$$\simeq \begin{pmatrix} \sum_{p=1}^{N_T} w^p & \sum_{p=1}^{N_T} \theta_z^p w^p & \frac{q \sum_{p=1}^{N_T} t_x^p w^p}{d} \\ -\sum_{p=1}^{N_T} \theta_z^p w^p & \sum_{p=1}^{N_T} w^p & \frac{q \sum_{p=1}^{N_T} t_y^p w^p}{d} \\ 0 & 0 & 1 \end{pmatrix}_x \mathbf{x} \quad (\text{S23})$$

$$= \begin{pmatrix} 1 & \theta_z^c & \frac{qt_x^c}{d} \\ -\theta_z^c & 1 & \frac{qt_y^c}{d} \\ 0 & 0 & 1 \end{pmatrix}_x \mathbf{x} \quad (\text{S24})$$

$$\simeq \begin{pmatrix} \cos(\theta_z^c) & \sin(\theta_z^c) & \frac{qt_x^c}{d} \\ -\sin(\theta_z^c) & \cos(\theta_z^c) & \frac{qt_y^c}{d} \\ 0 & 0 & 1 \end{pmatrix}_x \mathbf{x} = H_x^c \mathbf{x} \quad (\text{S25})$$

where H^c is the homography corresponding to the centroid of MDF. The equivalence in Eq. (S20) is valid since the homography warp is a linear transform for the considered 3D approximation and w^p is a scalar value. From Eq. (S19) and Eq. (S25) we obtain the following desired result.

$$u_c = H_x^c \mathbf{x} - x \quad (\text{S26})$$

A similar result can be shown for the case of v_c too, proving that the centroid of MDF will correspond to the centroid of PSF, if in-plane rotation of camera is small.

Discussion: In the approximation of the term $\sum_{p=1}^{N_T} \cos(\theta_z^p) w^p$ with $\cos(\sum_{p=1}^{N_T} \theta_z^p w^p)$ that we employ in the proof for Claim 1, the resulting error in centroid depends both on the spatial location in the image as well camera pose weights. The maximum error in alignment will result when pose weights are non-zero for extreme poses and are equal in magnitude. Previous studies ([2, 8]) have shown that natural camera shake contains rotations less than 5 degrees. Here we analyze the error in centroid (under the worst-case scenario) that will be introduced by the approximation in our claim. Assuming camera rotation up to 5 degrees, $w^{\theta=0} = w^{\theta=5} = 0.5$ and 0 elsewhere (since $\sum_{p=1}^{N_T} w^p = 1$), the error in centroid (induced by $\cos(\theta)$ approximation) at the border pixel (i.e., pixel at $y = 1000$, assuming an image of resolution of 2000×2000 with the origin at the center of the image) will be 0.951 which will increase to 0.992 if the error introduced by $\sin(\theta)$ is also taken into account. The blur kernel length for this case will be around 90 pixels which signifies the fact that the centroid error is relatively very small. Fig. 2 in the main paper shows our preliminary analysis (for image size 690×460) on the error behavior in CME while using the proposed centroid alignment. To address the case of high-resolution images, we repeated the analysis for the extreme case of $w^{\theta=0} = w^{\theta=5} = 0.5$, and image resolution of 2000×2000 . The NCC value (please refer to Section 4.1 in the main paper for details) for CME using edge profiles far-away from center was found to be 0.996 whereas the same with those near to the center was 0.9993. This shows that the centroid error has negligible impact on CME. In practice, one can reduce this effect further by giving more

weights to the constraint derived from edge profiles close to the center.

S2.3. Proof for claim 2

Claim 2: The centroid aligned PSFs correspond to a camera motion and a latent image both of which are consistent with the given blurred image and correct upto a homography.

Proof: Let us consider a collection of m PSFs k_1, \dots, k_m obtained from locations $\mathbf{x}_1, \dots, \mathbf{x}_m$ in the blurred image. What we already know is that the centroid of all these m PSFs corresponds to H^c . i.e.,

$$\mathbf{u}_{k_i}^c + \mathbf{x}_i = \mathbf{x}_{k_i}^c = H^c \mathbf{x}_i \quad (\text{S27})$$

where $\mathbf{u}_{k_i}^c$ is the centroid of PSF k_i and $\mathbf{x}_{k_i}^c$ is the corresponding image-coordinate. Let there are n active (with non-zero weights) poses in the MDF w_0 , with H^1, \dots, H^n being the corresponding homographies. Similarly, any point $(\mathbf{u}_{k_i}^j)$ in the blur kernel k_i is related to the kernel location \mathbf{x}_i through the corresponding homography H^j as

$$\mathbf{u}_{k_i}^j + \mathbf{x}_i = \mathbf{x}_{k_i}^j = H^j \mathbf{x}_i \quad (\text{S28})$$

Combining Eq. (S27) and Eq. (S28) we get the following relation,

$$\mathbf{x}_{k_i}^j = H^j(H^c)^{-1} \mathbf{x}_{k_i}^c \quad (\text{S29})$$

Without loss of generality, let us assume that we will align the PSFs by shifting their centroid to $(0,0)$, i.e., points in the PSF will get shifted along X and Y by the current PSF centroid $\mathbf{u}_{k_i}^c$. Therefore, in the centroid aligned PSF, the point $\mathbf{u}_{k_i}^j$ will get shifted to $\mathbf{u}_{k_i}^j - \mathbf{u}_{k_i}^c$. Using Eq. (S27), Eq. (S28), and Eq. (S29) we obtain the following results.

$$\mathbf{u}_{k_i}^j - \mathbf{u}_{k_i}^c = H^j(H^c)^{-1} \mathbf{x}_{k_i}^c - \mathbf{x}_i - \mathbf{u}_{k_i}^c \quad (\text{S30})$$

$$= H^j(H^c)^{-1} \mathbf{x}_{k_i}^c - \mathbf{x}_{k_i}^c \quad (\text{S31})$$

This shows that the centroid aligned PSFs corresponds to a different MDF w_0^c (formed of $H^j(H^c)^{-1}$) and latent image L^c (defined on the coordinates $\mathbf{x}_{k_i}^c$) both of which are consistent with the same blurred image.

S2.4. Proof for claim 3

Claim 3: The absolute value of differential of edge profile is equivalent to the absolute gradient of a blurred image (at the location of the edge profile) normalized by the difference between the two homogeneous colors involved.

Proof: The equivalence in claim 3 is illustrated in Fig. S1. Consider two adjacent pixels along a line L over which an edge profile exists in the image. We can express them in terms of the corresponding colors from homogeneous regions (similar to Eq. (1)) as

$$L(i) = \alpha_i c_1 + (1 - \alpha_i) c_2 \quad (\text{S32})$$

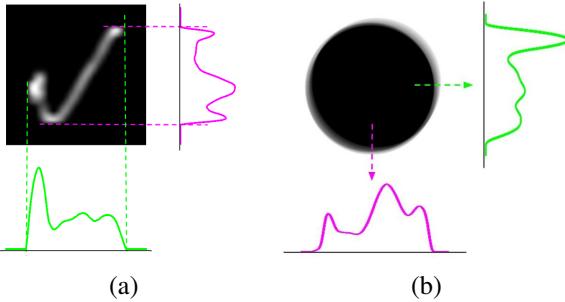


Figure S1. (a) Example to show that extreme points of edge profiles at different directions need not correspond to the same homographies, the plot showing in green and magenta are the value of differential of edge profile obtained by projecting PSF onto X and Y axis respectively. (b) The gradient along the direction of edge profile extracted from a corresponding blurred image (The equivalence between (a) and (b) is discussed in claim 3 and its proof).

$$L(i+1) = \alpha_{i+1}c_1 + (1 - \alpha_{i+1})c_2 \quad (\text{S33})$$

The first-order gradient of L at i is given by

$$\nabla L = L(i+1) - L(i) \quad (\text{S34})$$

$$= (\alpha_{i+1} - \alpha_i)c_1 - (\alpha_{i+1} - \alpha_i)c_2 \quad (\text{S35})$$

$$= (c_1 - c_2)\nabla\alpha \quad (\text{S36})$$

where $(\nabla\alpha)$ is the gradient of edge profile. Eq. (S36) implies that the gradient of edge profiles is equivalent to the normalized form of the first-order gradient of B_I taken along the direction of the edge profile. From Eq. (4), Eq. (5), Eq. (6), and Eq. (S36) we can see that

$$\nabla B_I = (c_1 - c_2)P \quad (\text{S37})$$

i.e., gradient of blurred image along the direction of edge profile is equivalent to the projection of PSF onto the line along the same direction scaled by $(c_1 - c_2)$. Further, in Eq. (S37) we can replace ∇B_I and $(c_1 - c_2)$ with their absolute values (to form an equivalent condition), since the signs of both terms will cancel each other for the two possible scenarios ($c_1 < c_2$) and ($c_1 > c_2$). Thus

$$|\nabla B_I| = |(c_1 - c_2)|P \quad (\text{S38})$$

S3. Implementation details

In our experiments, we use the 3D trajectory approximation of $model_2$. To improve the speed of camera motion estimation by taking advantage of computations in scale space, we constrain the motion estimation only over a set of relevant poses for subsequent iterations. While moving across the scale we interpolate the MDF to find initial kernel estimate for new scale. The presence of our new constraint rules-out the possibility of getting stuck in local minima, since it directly relates step edges in latent image onto the

edge profiles in the blurred image. Nevertheless, it is better to perform the camera motion estimation in a scale space fashion to speed up the entire process by estimating only over a constrained set from the whole camera pose space.

The parameter settings used for obtaining the results for our proposed method are the following. We have used $\lambda = 0.01$ for the proposed method **EpAlone**. For **Xu + Ep**, we use $\lambda_f = 0.003$, $\lambda_{I_p} = 1$, $\lambda_w = 0.01$, and $n_{max} = 5$.

S4. Large blur kernels

Large blur kernels of size more than 50 pixels are uncommon in practice. For such cases, the improvement of our approach over [10] is expected to be marginal, since the edge profile constraint will contain only few edges. Nevertheless, for completeness, we have provided performance comparison over the full dataset of [5] in Fig. S2. As can be observed, the inclusion of large blur kernels leads to only a small reduction in PSNR improvement for our method. In Fig. S3, we have provided one example for large blur kernel to illustrate the fact that our result is qualitatively comparable to that of [10].

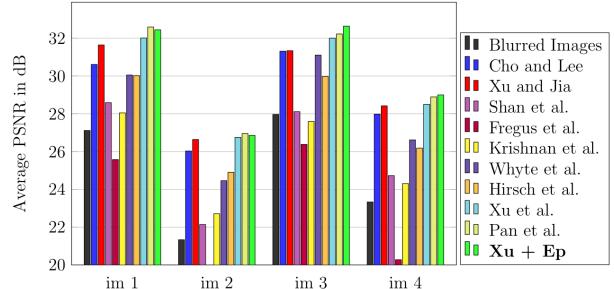


Figure S2. Quantitative evaluation on all kernels from [5].

S5. Running time

In Table S1, we list the running times of our implementation for space invariant kernel estimation. Since **EpAlone** does not have explicit edge prediction stages and alternating minimizations, kernel estimation is much faster as compared to other methods. As can be observed, our novel way of incorporating the proposed constraint reduces the computational complexity of our proposed method **EpAlone** over its space invariant counterpart proposed in [1]. Also the computational complexity of the state-of-the-art method in [6] which performs equally well as compared to **Xu + Ep**, is significantly high as compared to **Xu + Ep**.

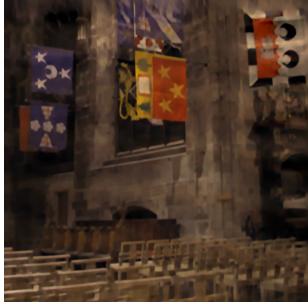
In Table S2 we list the running times of our proposed approaches for performing space variant CME. Again, **EpAlone** is more efficient than other approaches. As is evident from Table S2, **Xu + Ep** is able to perform non-uniform deblurring with only a marginal increase in computational complexity over [10]. Running time reported here is



(a) Motion blurred image



(b) Xu et al. [9]



(c) Pan et al. [6]



(d) **Xu + Ep**

Figure S3. Example image from the benchmark dataset [5], corresponding to large blur kernel.

obtained using Matlab implementation of all the algorithms, and on a PC with Intel i5 CPU and 16GB memory.

Table S1. Run time (in seconds) for the proposed methods and existing methods on SI deblurring for an image of size 800×800 and kernel of size 27×27

| EpAlone | Xu + Ep | Cho [1] | Xu [10] | Pan [6] |
|----------------|----------------|---------|---------|---------|
| 17.32 | 35.63 | 80.99 | 22.02 | 1564.78 |

Table S2. Run time (in seconds) for the proposed methods for non-uniform deblurring

| Image size | Kernel size | EpAlone | Xu + Ep | Xu [10] |
|------------------|----------------|----------------|----------------|---------|
| 366×274 | 11×11 | 29.4 | 84.05 | 77.15 |
| 768×512 | 9×9 | 45.9 | 170.95 | 152.99 |
| 768×512 | 23×23 | 74.2 | 296.2 | 286.91 |

S6. Additional results

In this section, we give performance comparison over more examples from real image datasets of existing works.

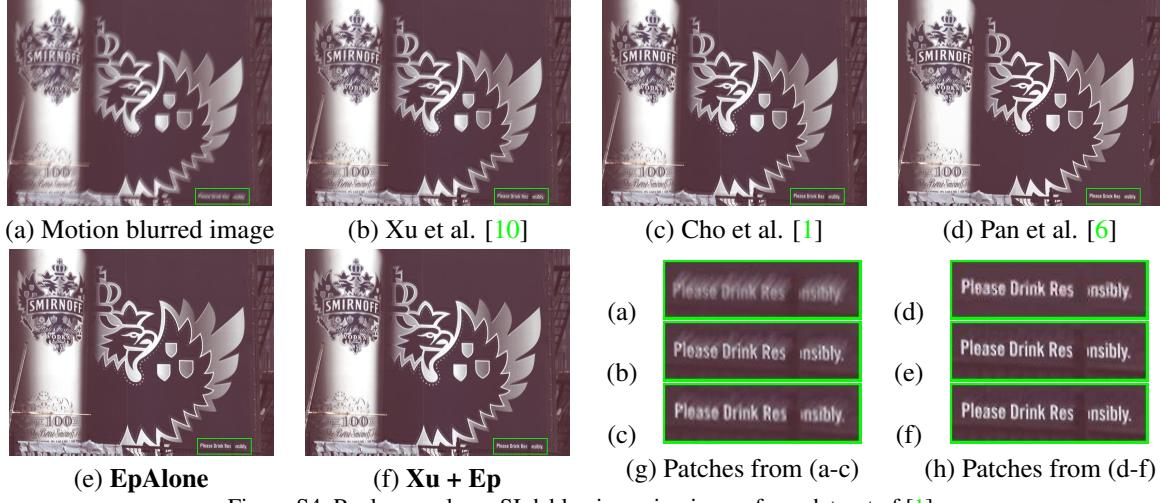


Figure S4. Real example on SI deblurring using image from dataset of [1].

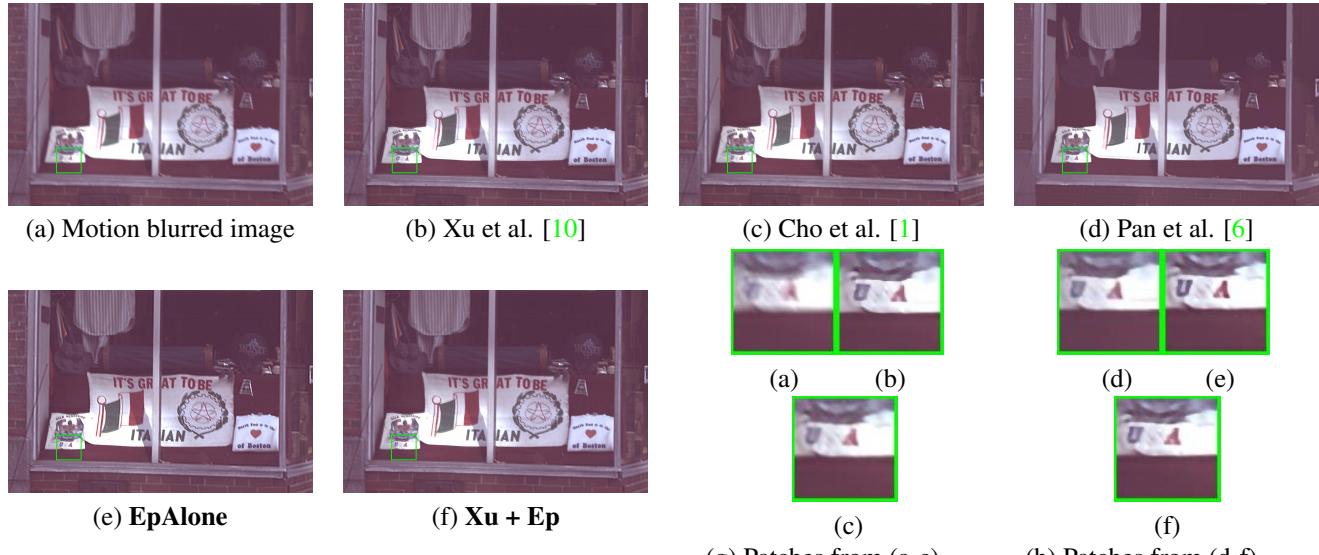


Figure S5. Real example on SI deblurring using image from dataset of Cho et al. [1].



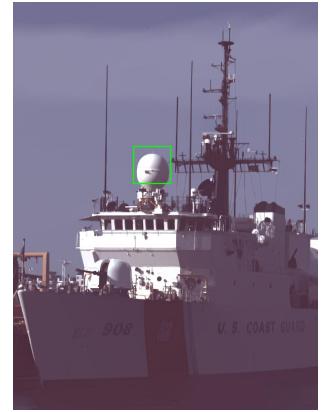
(a) Motion blurred image



(b) Xu et al. [10]



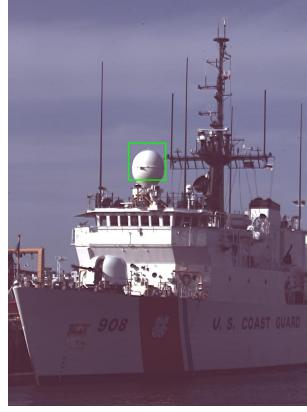
(c) Cho et al. [1]



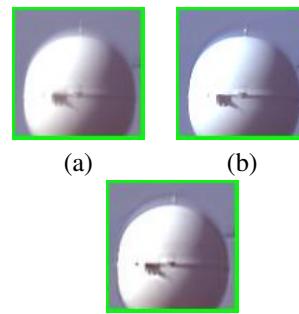
(d) Pan et al. [6]



(e) EpAlone



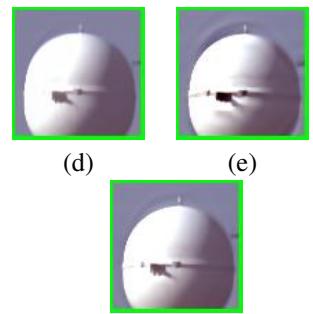
(f) Xu + Ep



(a) (b)



(c)



(d) (e)



(f)

(g) Patches from (a-c)

(h) Patches from (d-f)

Figure S6. Real example on SI deblurring using image from dataset of Cho et al. [1].



(a) Motion blurred image



(b) Xu et al. [10]



(c) Cho et al. [1]



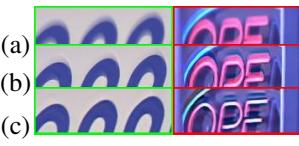
(d) Pan et al. [6]



(e) EpAlone



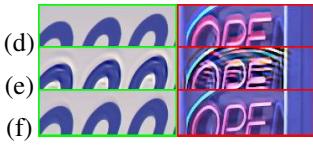
(f) Xu + Ep



(a) (b)



(g) Patches from (a-c)

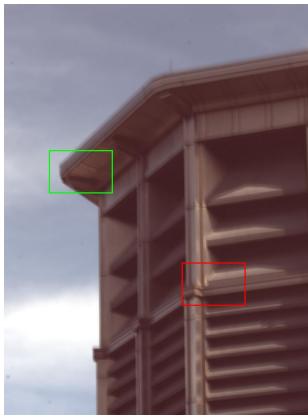


(d) (e)

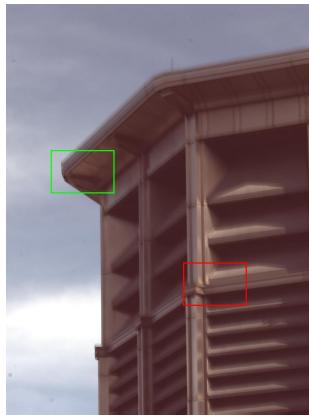


(h) Patches from (d-f)

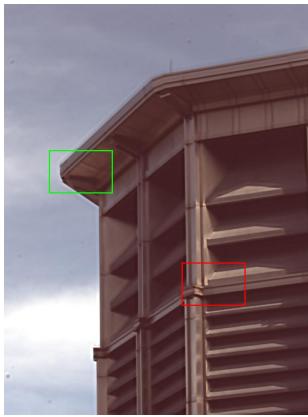
Figure S7. Real example on SI deblurring using image from dataset of Cho et al. [1].



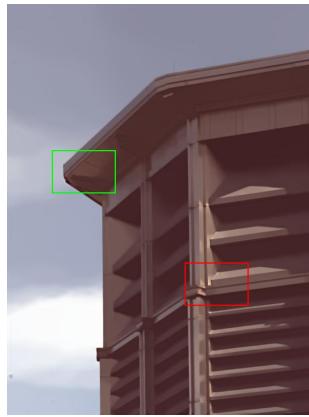
(a) Motion blurred image



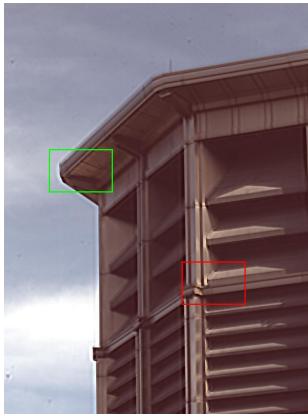
(b) Xu et al. [10]



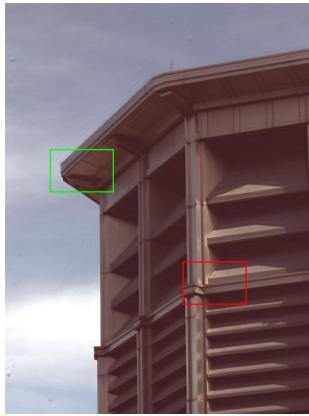
(c) Cho et al. [1]



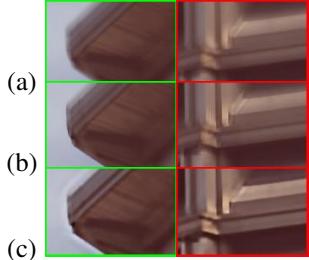
(d) Pan et al. [6]



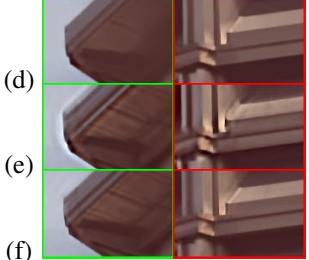
(e) EpAlone



(f) Xu + Ep

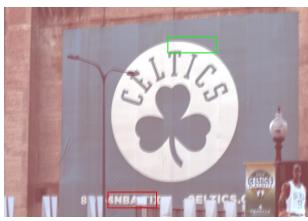


(g) Patches from (a-c)



(h) Patches from (d-f)

Figure S8. Real example on SI deblurring using image from dataset of [1].



(a) Motion blurred image



(b) Xu et al. [10]



(c) Cho et al. [1]



(d) Pan et al. [6]



(e) EpAlone



(f) Xu + Ep



(g) Patches from (a-c)



(h) Patches from (d-f)

Figure S9. Real example on SI deblurring using image from dataset of Cho et al. [1].

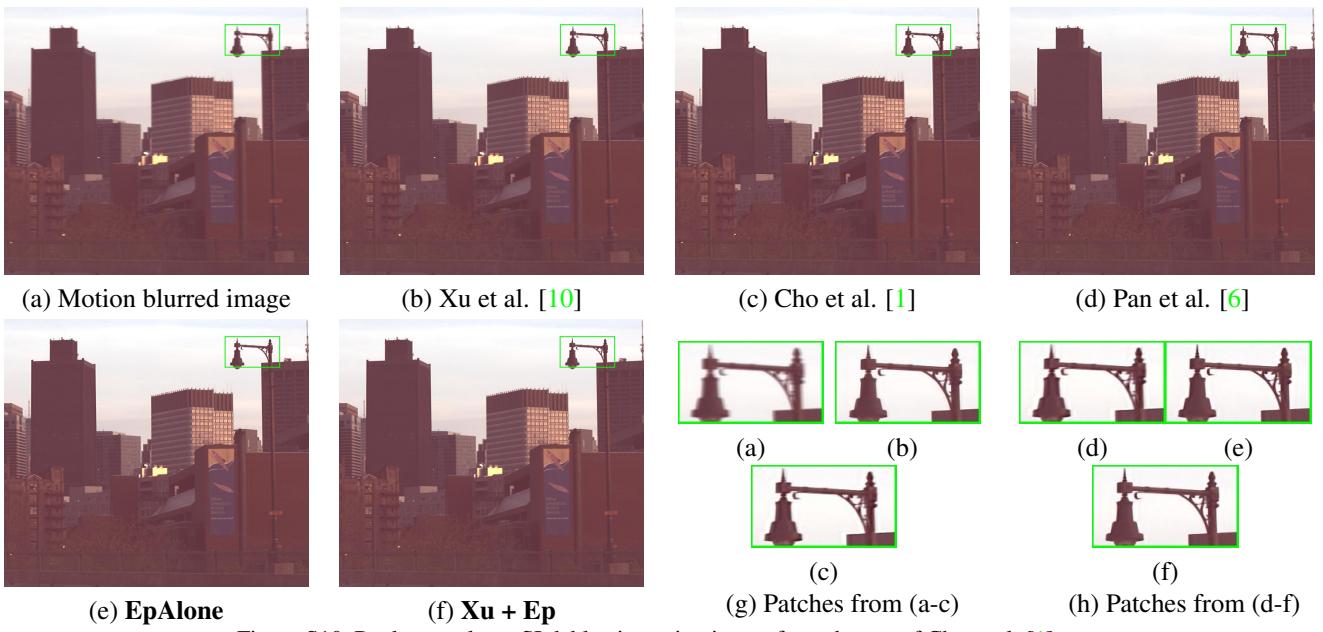


Figure S10. Real example on SI deblurring using image from dataset of Cho et al. [1].

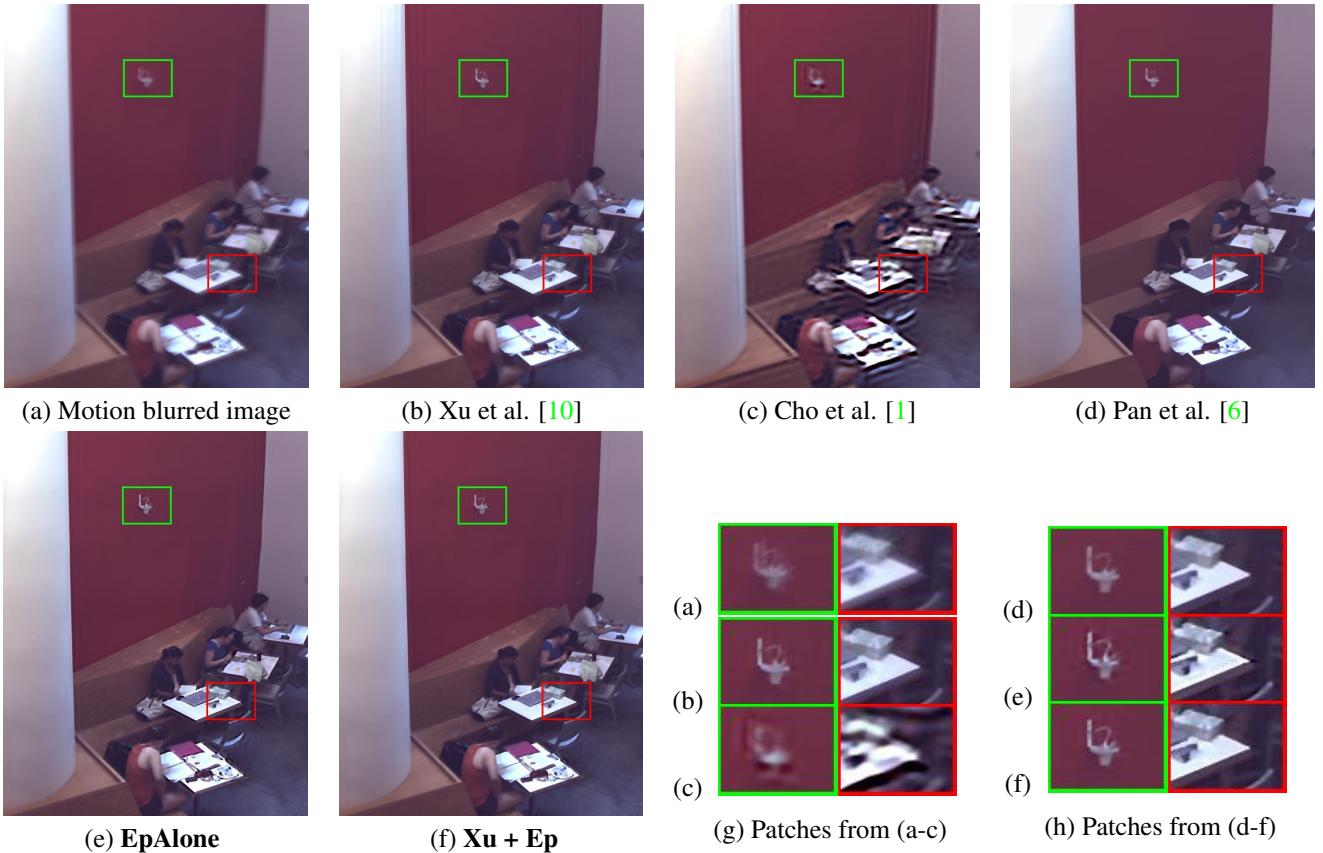


Figure S11. Real example on SI deblurring using image from dataset of [1].

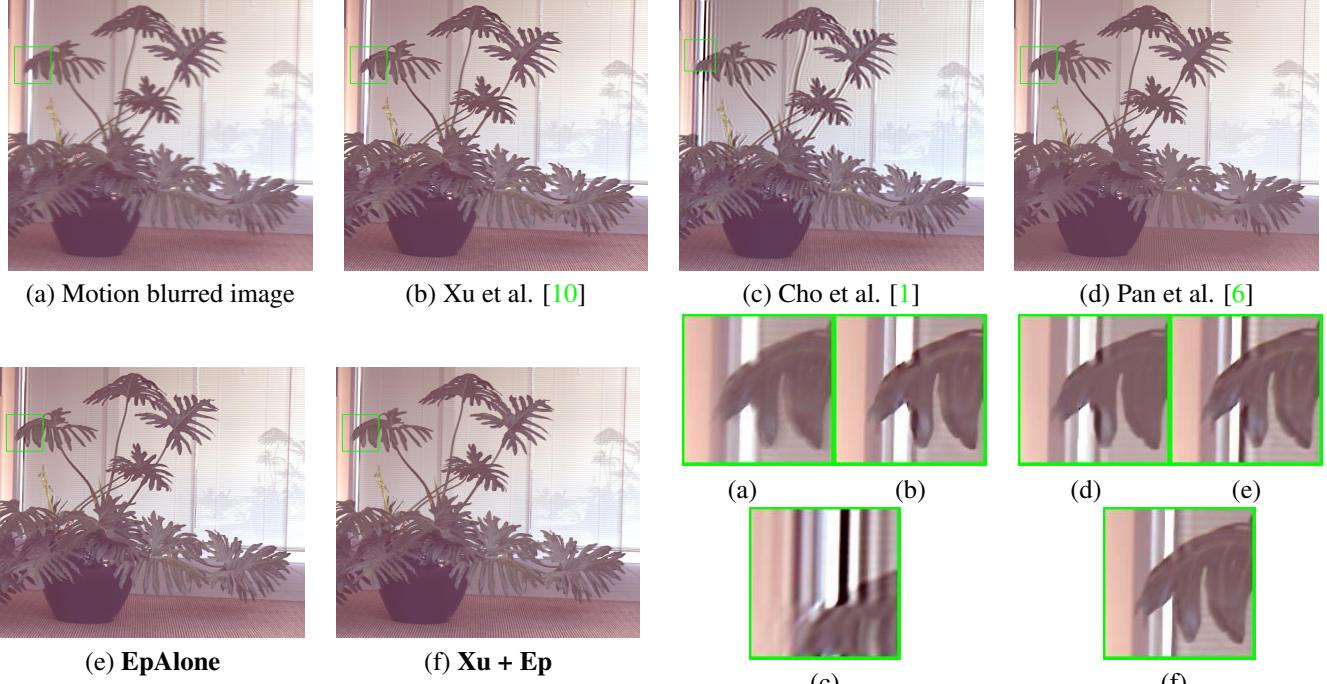


Figure S12. Real example on SI deblurring using image from dataset of Cho et al. [1].

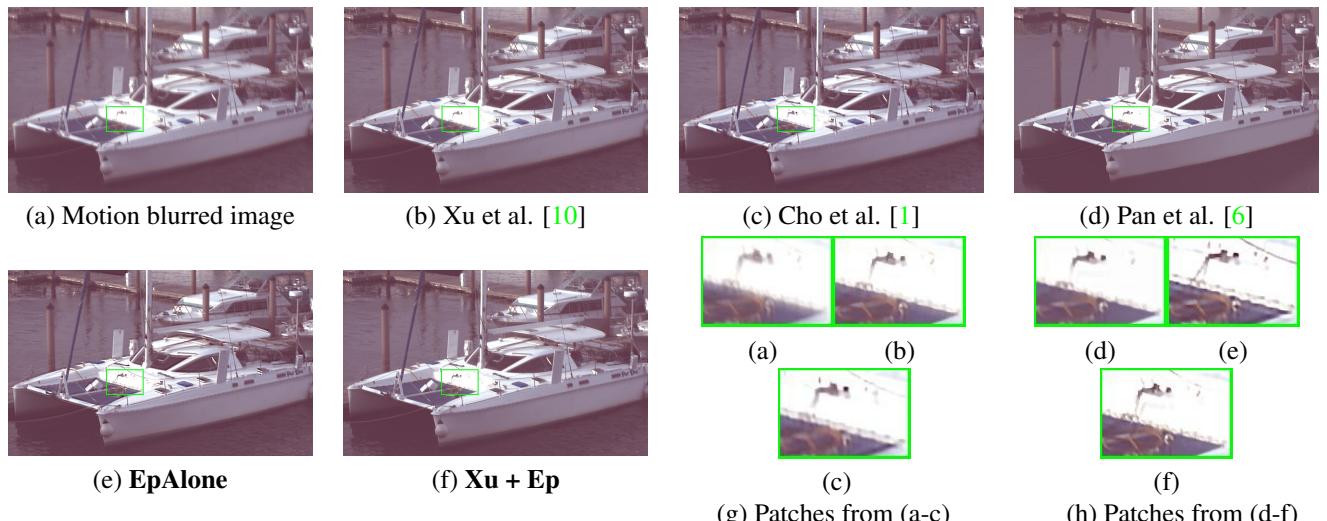


Figure S13. Real example on SI deblurring using image from dataset of [1].

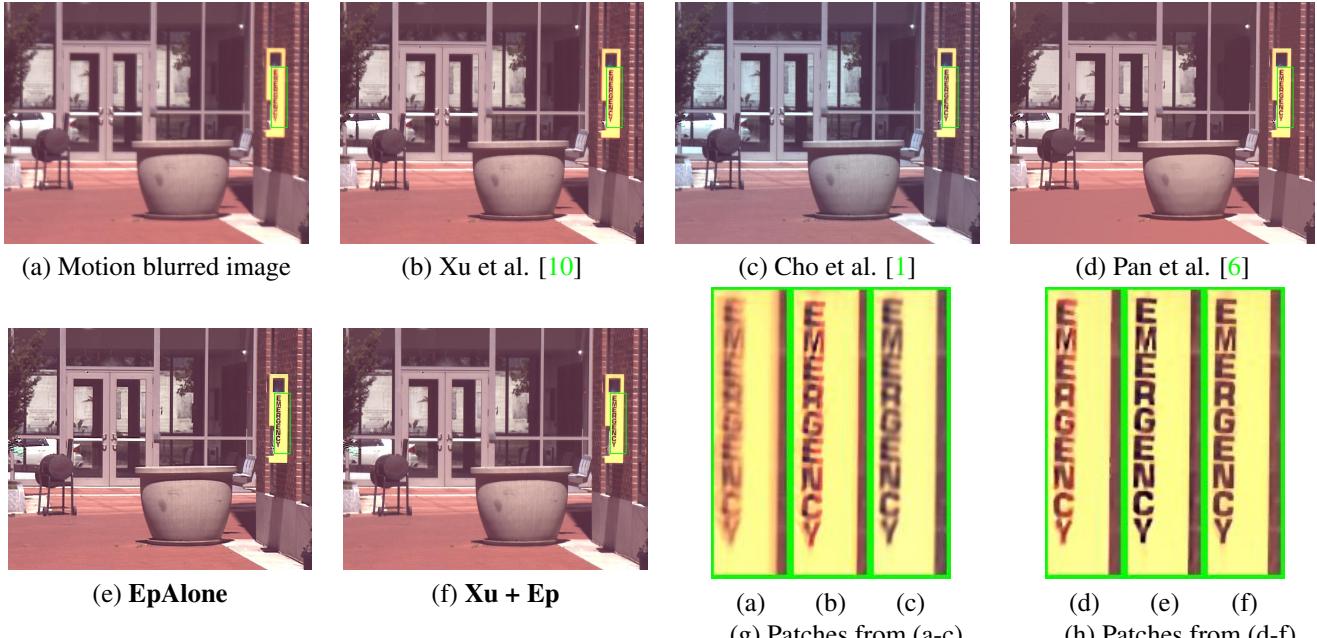


Figure S14. Real example on SI deblurring using image from dataset of [1].

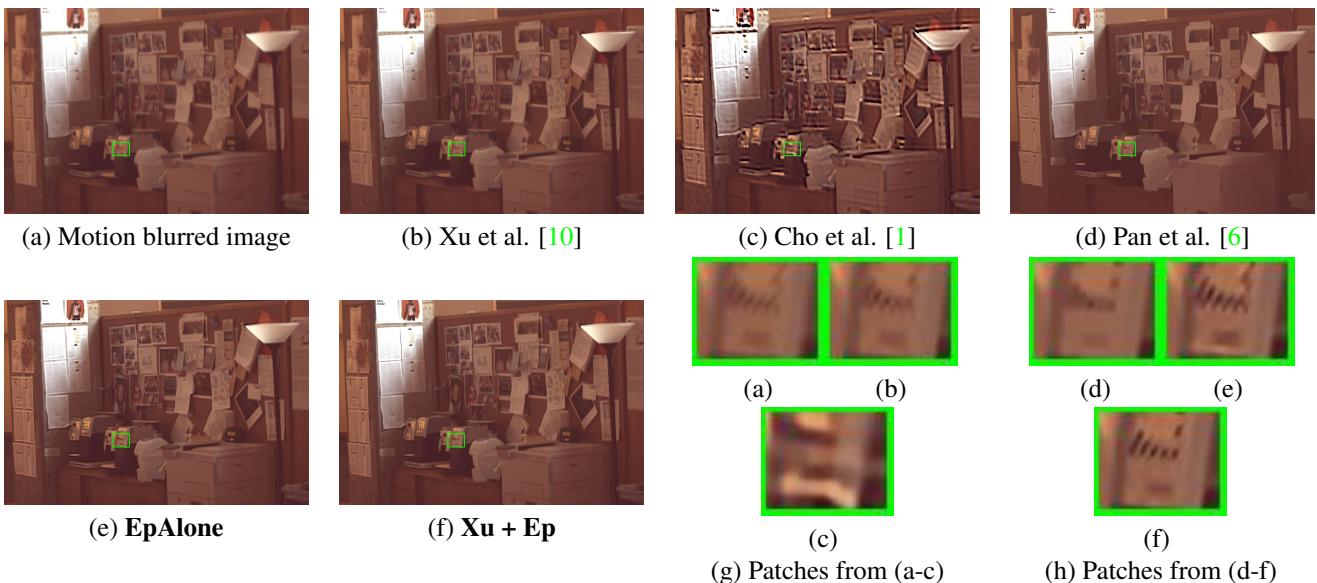


Figure S15. Real example on SI deblurring using image from dataset of [1].

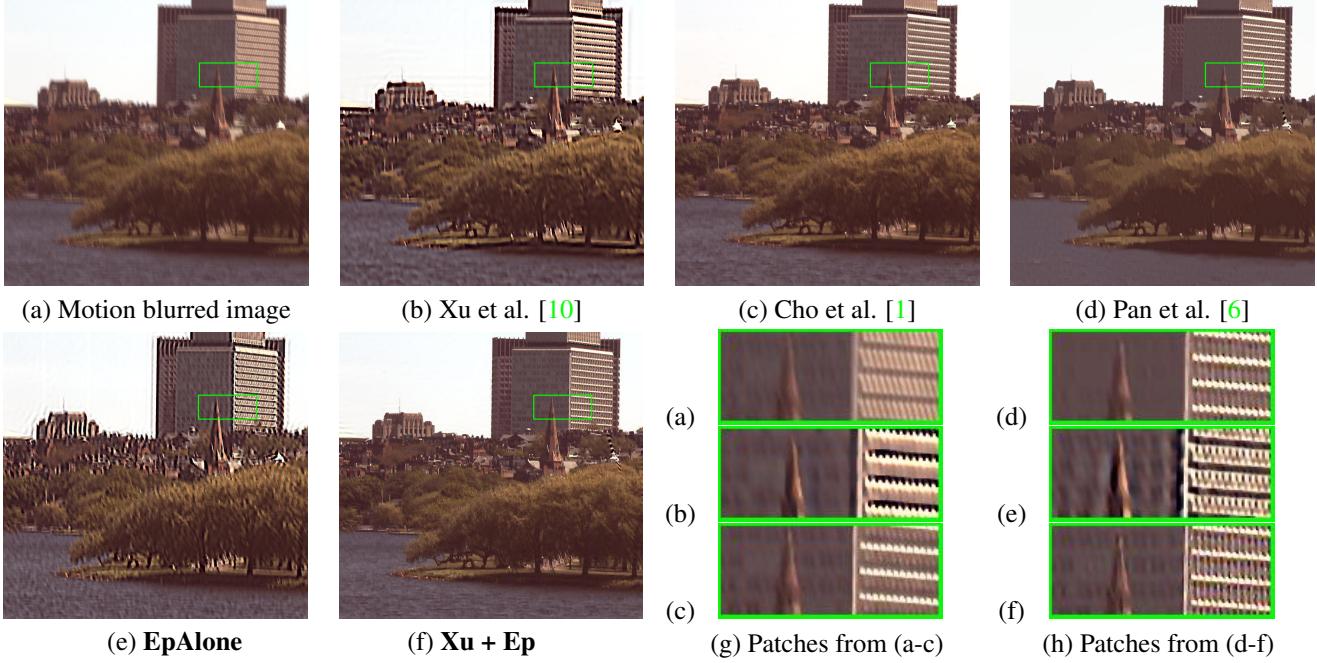


Figure S16. Real example on SI deblurring using image from dataset of Cho et al. [1].

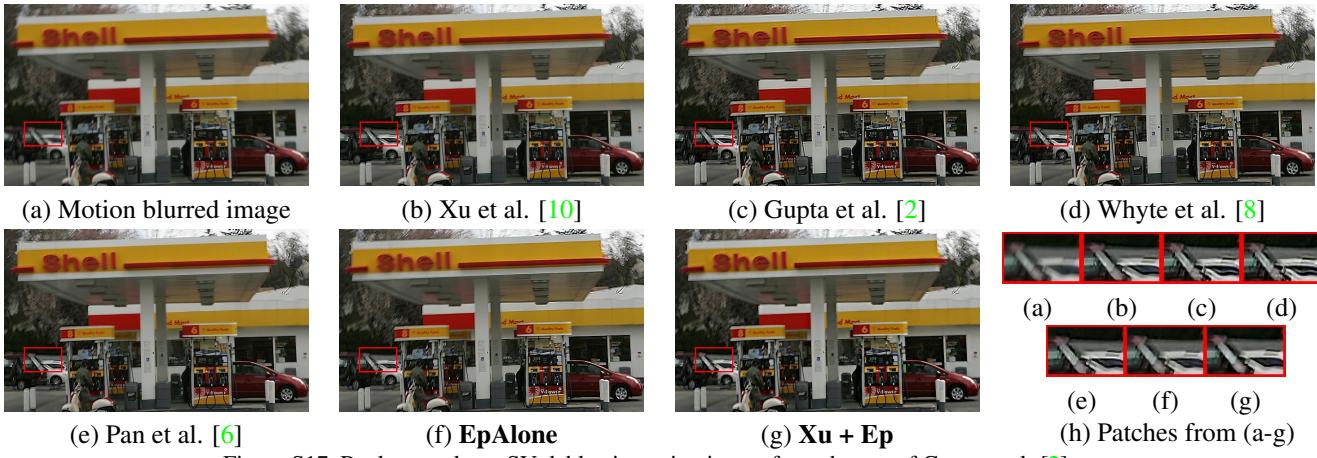


Figure S17. Real example on SV deblurring using image from dataset of Gupta et al. [2].

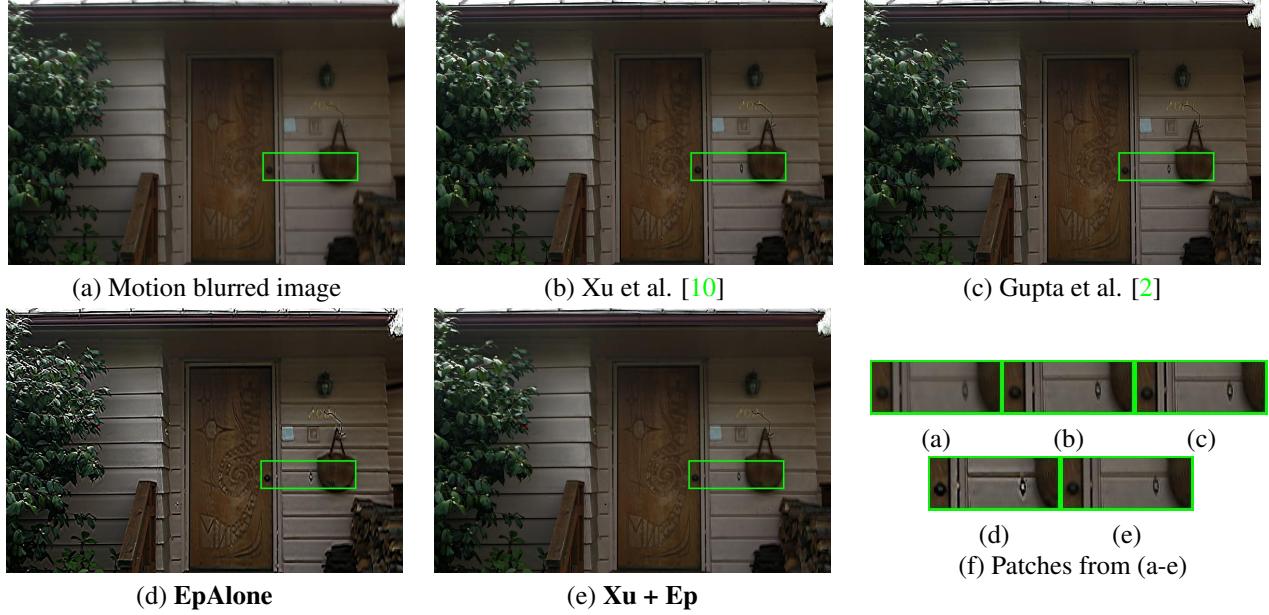


Figure S18. Real example on SV deblurring using image from dataset of Gupta et al. [2].

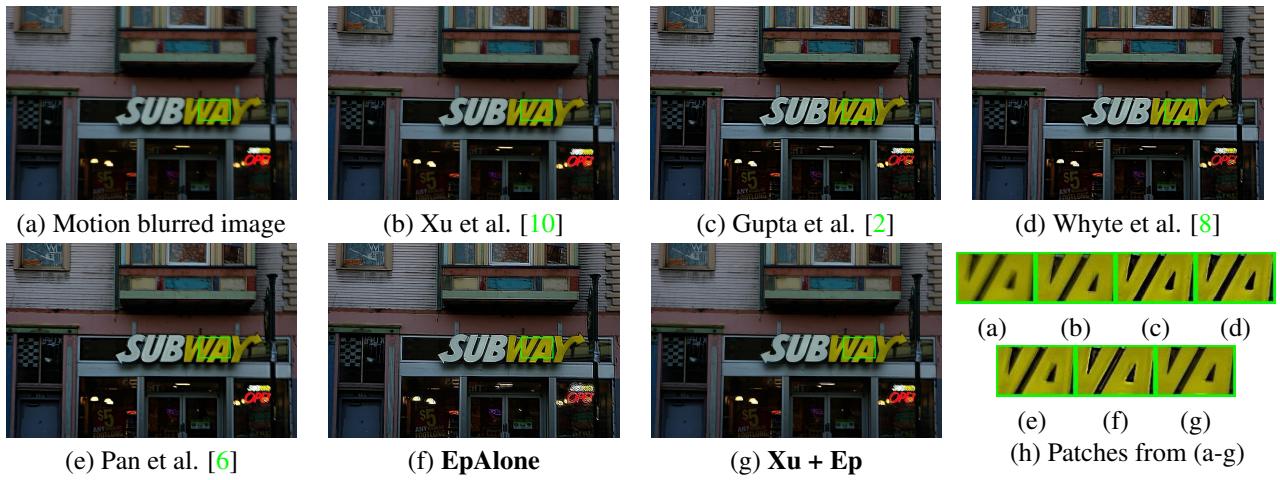


Figure S19. Real example on SV deblurring using image from dataset of Gupta et al. [2].

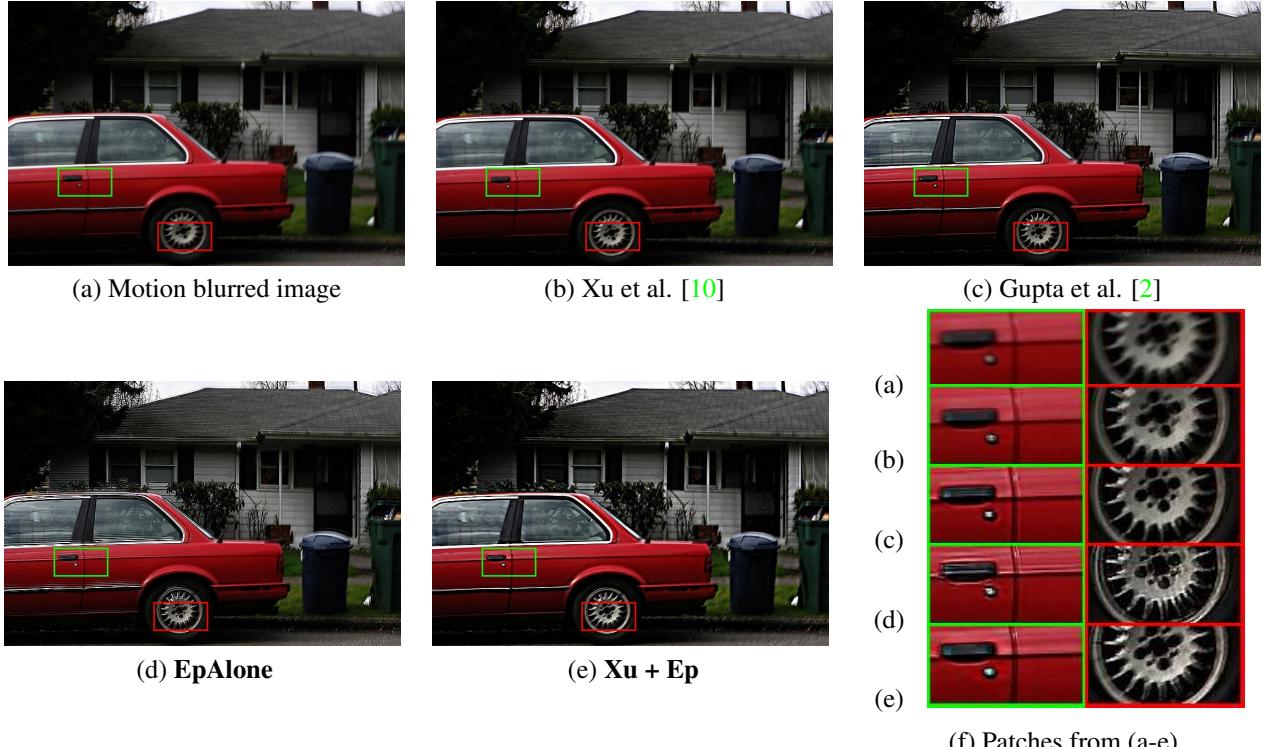


Figure S20. Real example on SV deblurring using image from dataset of Gupta et al. [2].

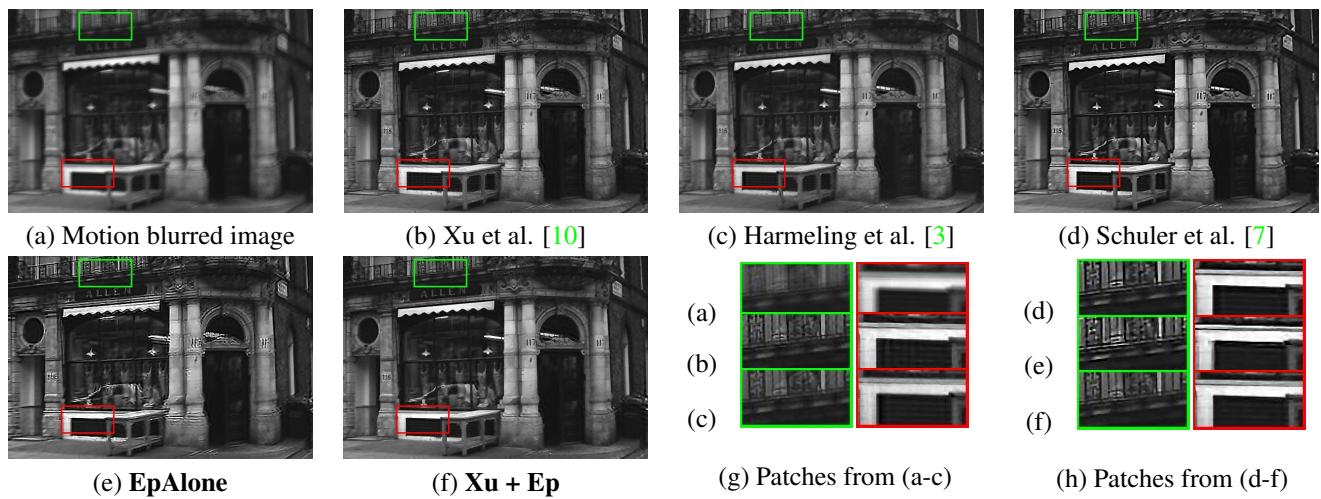


Figure S21. Real example on SV deblurring using image from dataset of Harmeling et al. [3].

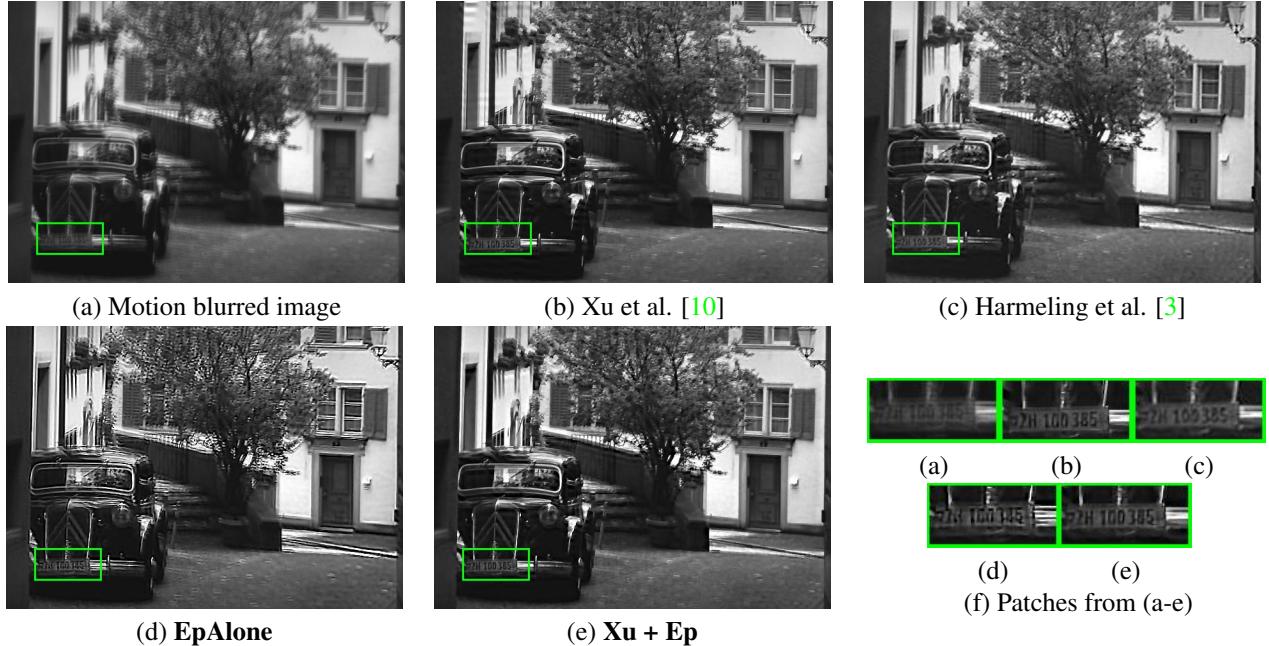


Figure S22. Real example on SV deblurring using image from dataset of Harmeling et al. [3].

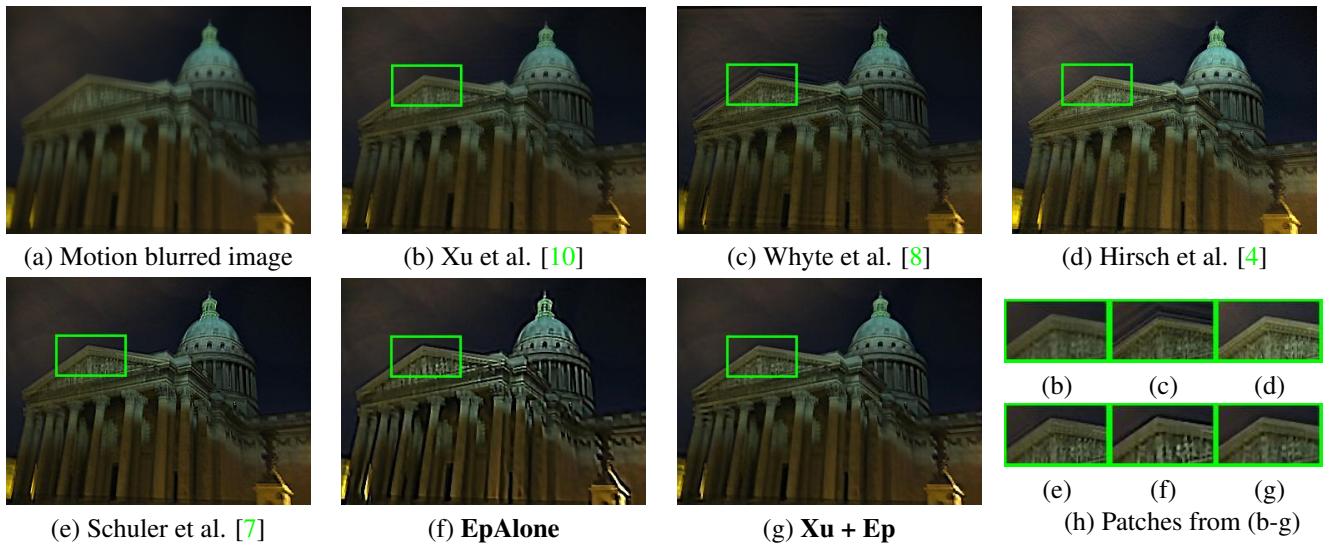


Figure S23. Real example on SV deblurring using image from dataset of Whyte et al. [8].

References

- [1] T. S. Cho, S. Paris, B. K. Horn, and W. T. Freeman. Blur kernel estimation using the radon transform. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 241–248. IEEE, 2011. [4](#), [5](#), [6](#), [7](#), [8](#), [9](#), [10](#), [11](#), [12](#)
- [2] A. Gupta, N. Joshi, C. L. Zitnick, M. Cohen, and B. Curless. Single image deblurring using motion density functions. In *European Conference on Computer Vision*, pages 171–184. Springer, 2010. [3](#), [12](#), [13](#), [14](#)
- [3] S. Harmeling, H. Michael, and B. Schölkopf. Space-variant single-image blind deconvolution for removing camera shake. In *Advances in Neural Information Processing Systems*, pages 829–837, 2010. [14](#), [15](#)
- [4] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Schölkopf. Fast removal of non-uniform camera shake. In *2011 International Conference on Computer Vision*, pages 463–470. IEEE, 2011. [15](#)
- [5] R. Köhler, M. Hirsch, B. Mohler, B. Schölkopf, and S. Harmeling. Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database. In *European Conference on Computer Vision*, pages 27–40. Springer, 2012. [4](#), [5](#)
- [6] J. Pan, D. Sun, H. Pfister, and M.-H. Yang. Blind image deblurring using dark channel prior. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. [4](#), [5](#), [6](#), [7](#), [8](#), [9](#), [10](#), [11](#), [12](#), [13](#)
- [7] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schölkopf. Learning to deblur. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(7):1439–1451, 2016. [14](#), [15](#)
- [8] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010. [3](#), [12](#), [13](#), [15](#)
- [9] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *European conference on computer vision*, pages 157–170. Springer, 2010. [5](#)
- [10] L. Xu, S. Zheng, and J. Jia. Unnatural 10 sparse representation for natural image deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1107–1114, 2013. [4](#), [5](#), [6](#), [7](#), [8](#), [9](#), [10](#), [11](#), [12](#), [13](#), [14](#), [15](#)