# Revisiting Cross-channel Information Transfer for Chromatic Aberration Correction

Tiancheng Sun[2,1] Yifan Peng[3,1] Wolfgang Heidrich[1,3]
[1]King Abdullah University of Science and Technology, Thuwal, Saudi Arabia
[2]IIIS, Tsinghua University, Beijing, China
[3]The University of British Columbia, Vancouver, Canada

## 1. Algorithm analysis

### 1.1. Flow chart

We provide a detailed flow chart of our joint algorithm. As shown in Fig. 1, we first make a guess of latent image using the proposed initialization, then iterate between PSF estimation and cross-channel transfer (CCT) step until reaching the convergence or termination condition.

During the iterating process, we apply a simple trick to avoid boundary effects of convolution. Notice that in the main text, step 6 of the algorithm is written in a slightly different way as:

$$\boldsymbol{\alpha} = \underset{\boldsymbol{\alpha}}{\text{argmin}} \left\| \mathbf{J}_{\mathrm{s}}^{\mathrm{CCT}} - \mathbf{B}_{\mathrm{s}} * (\mathbf{T}(\mathbf{I}_{\mathrm{G}}^{\mathrm{CCT}}) \cdot \boldsymbol{\alpha}) \right\|_2^2. \tag{1}$$

In practice, we put the convolution operation inside $\mathbf{T}$ since both of them are linear. However, operating convolution on each CCT window may introduce artifacts at the borders. In the contrast, we pre-compute the full size image of $\mathbf{B}_{\mathrm{s}} * \mathbf{I}_{\mathrm{G}}$ in the PSF estimation step, and crop the CCT windows from the pre-computed image in the CCT step, as shown in Fig. 1.

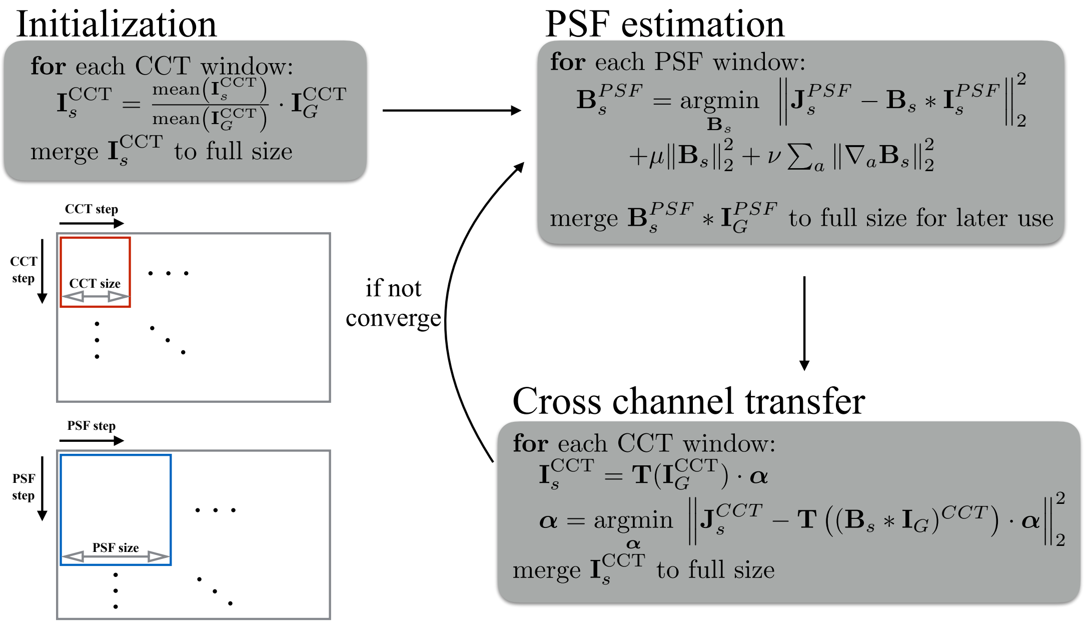

Initialization
**for** each CCT window:
$\mathbf{I}_s^{\mathrm{CCT}} = \frac{\mathrm{mean}(\mathbf{I}_s^{\mathrm{CCT}})}{\mathrm{mean}(\mathbf{I}_G^{\mathrm{CCT}})} \cdot \mathbf{I}_G^{\mathrm{CCT}}$
merge $\mathbf{I}_s^{\mathrm{CCT}}$ to full size
PSF estimation
**for** each PSF window:
$\mathbf{B}_s^{PSF} = \underset{\mathbf{B}_s}{\text{argmin}} \left\| \mathbf{J}_s^{PSF} - \mathbf{B}_s * \mathbf{I}_s^{PSF} \right\|_2^2 + \mu \|\mathbf{B}_s\|_2^2 + \nu \sum_a \|\nabla_a \mathbf{B}_s\|_2^2$
merge $\mathbf{B}_s^{PSF} * \mathbf{I}_G^{PSF}$ to full size for later use
CCT step
CCT step
CCT size
PSF step
PSF step
PSF size
if not converge
Cross channel transfer
**for** each CCT window:
$\mathbf{I}_s^{\mathrm{CCT}} = \mathbf{T}(\mathbf{I}_G^{\mathrm{CCT}}) \cdot \boldsymbol{\alpha}$
$\boldsymbol{\alpha} = \underset{\boldsymbol{\alpha}}{\text{argmin}} \left\| \mathbf{J}_s^{CCT} - \mathbf{T}\left((\mathbf{B}_s * \mathbf{I}_G)^{CCT}\right) \cdot \boldsymbol{\alpha} \right\|_2^2$
merge $\mathbf{I}_s^{\mathrm{CCT}}$ to full size

Figure 1: Working flow chart of our joint algorithm. The left two insets illustrate the different window selection for CCT step and PSF step.

## 1.2. Convergence analysis

Fig. 2 shows the SSIM assessments of images subject to iteration rounds. From the plots we see that the image quality converges very quickly after the first couple iterations (i.e. 3 iterations), and may drop a little bit when the iterating process continues. We ascribe this weak drop of image quality to possible over-fitting to the pixel values of green channel.

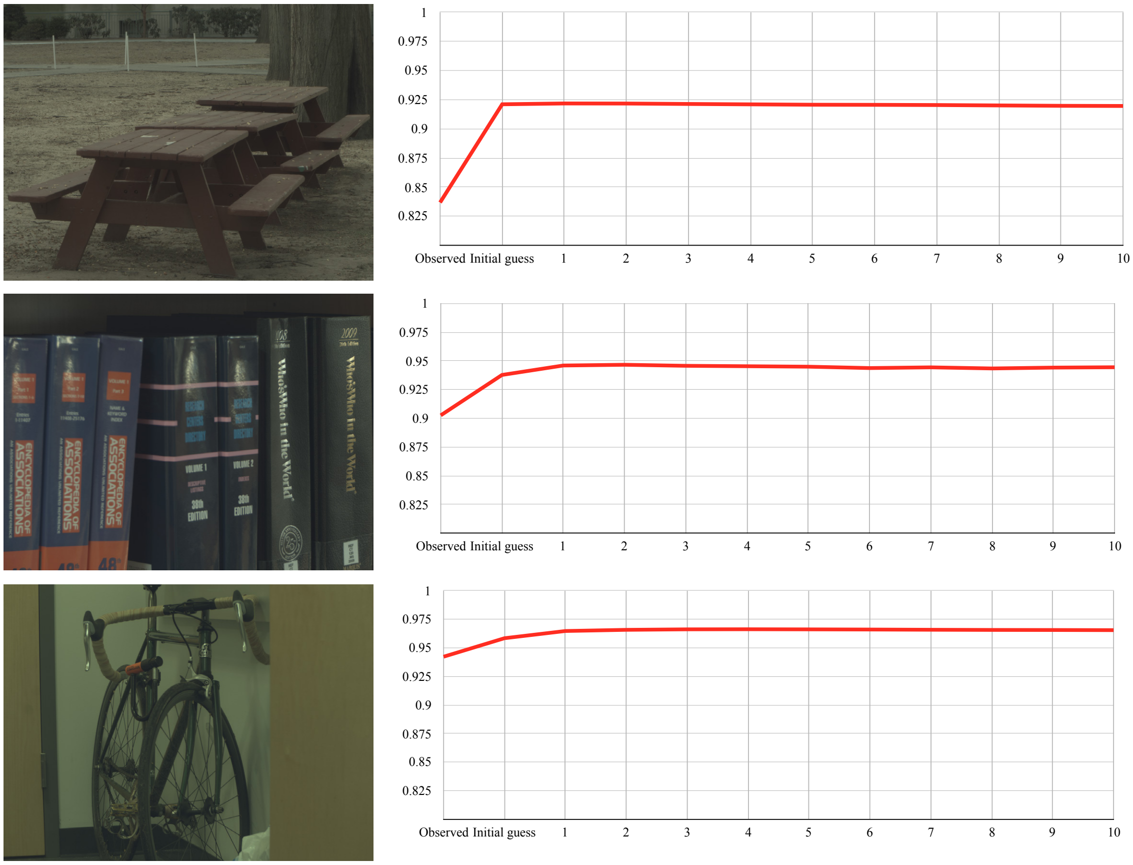


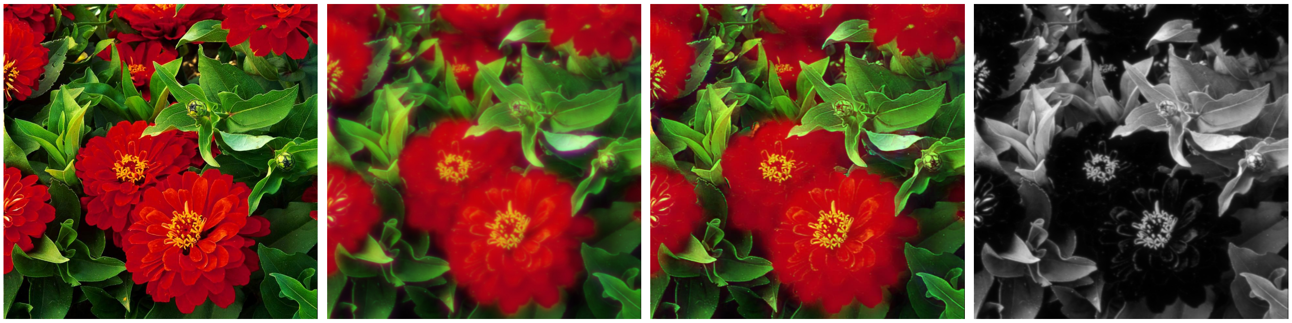
Figure 2: SSIM assessments of the restored images with different iteration rounds.

To examine the reasoning of over-fitting, we investigate an extreme case where the drastic hue change appears in the image, as shown in the left of Fig. 3. We synthetically blur the image using a blur kernel used in the main text, accordingly we have the blurred image (mid-left). Notice that in the green channel of this blurred image (presented in the right of Fig. 3), the pixel intensities are very low on the region of red flower. Apparently, the cross-channel correlation doesn't fit well into our model. In this scenario, directly using green channel as the reference channel may cause a significant loss of most fine details and the over-fitting in the plain black part of green channel. However, for other regions, including the stamen and some of the petals of red flower, the resolved image has preserved the details well, that in some sense validates the robustness of our algorithm.

The main reason accounting for this over-fitting is the simplicity of our cross- channel model. We yield this problem to future work by exploiting an improved model of cross-channel similarity, for instance using learning-based strategies.

## 1.3. Run time discussion

As reported in the main text, our algorithm takes around 7 secs to restore one channel of a 1,400 × 1,000 pixels image. It is worth noting that storing a whole image could take more than just 14 secs, considering the time (around 1 min) to blindly deconvolve [4] the reference channel. For most of the captured images, the green channel of the image is relatively sharp such that the deconvolution shall be fast and robust. Since we don't have the executable codes of state-of-the-art cross-channel deconvolution algorithms, it is tough to give an intuitive quantitative comparison. However, with the aid of the proposed compact but powerful image formation, we believe our algorithm is very competitive in terms of computational efficiency.



(a) ground truth image (b) blurred image (c) deblurred image (d) green channel (blurred)

Figure 3: Extreme example with drastic hue change (a possible failure case of our algorithm).

## 2. Results with an additional prior added on latent image

As mentioned in the main text, we have run the experiments with a gradient-based cross-channel prior added in the deconvolution scheme as a comparison. Specifically, we modify step 6 in the described algorithm into:

$$\boldsymbol{\alpha} = \underset{\boldsymbol{\alpha}}{\mathrm{argmin}} \ \left\|\mathbf{J}_{\mathrm{s}}^{\mathrm{CCT}} - \mathbf{B}_{\mathrm{s}} * \mathbf{I}_{\mathrm{s}}^{\mathrm{CCT}}\right\|_2^2 + \gamma\left\|\nabla\mathbf{I}_{\mathrm{s}}^{\mathrm{CCT}} \cdot \mathbf{I}_{\mathrm{G}}^{\mathrm{CCT}} - \nabla\mathbf{I}_{\mathrm{G}}^{\mathrm{CCT}} \cdot \mathbf{I}_{\mathrm{s}}^{\mathrm{CCT}}\right\|_2^2, \tag{2}$$

where

$$\mathbf{I}_{\mathrm{s}}^{\mathrm{CCT}} = \mathbf{T}(\mathbf{I}_{\mathrm{G}}^{\mathrm{CCT}}) \cdot \boldsymbol{\alpha}.$$

We found that the quality of the resulting images doesn't increase (even drop a little bit), as shown in Tab. 1. Although we are using $l2$ norm, the comparison result still suggests that our model has covered the function of a regular cross-channel prior. More specifically, we are directly using the pixel values of the reference channel, which fully satisfy the distribution of natural images, thus, we don't need further weak statistical priors to recover the corrupted details.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Average |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Without prior | 36.07 | 36.37 | 33.96 | 31.87 | 30.93 | 34.29 | 32.64 | 37.93 | 35.60 | 30.88 | 34.05 |
| With prior | 36.15 | 36.38 | 33.93 | 31.85 | 30.93 | 34.27 | 32.61 | 37.86 | 35.57 | 30.87 | 34.04 |

Table 1: Quantitative comparison on algorithm with and without gradient prior added.

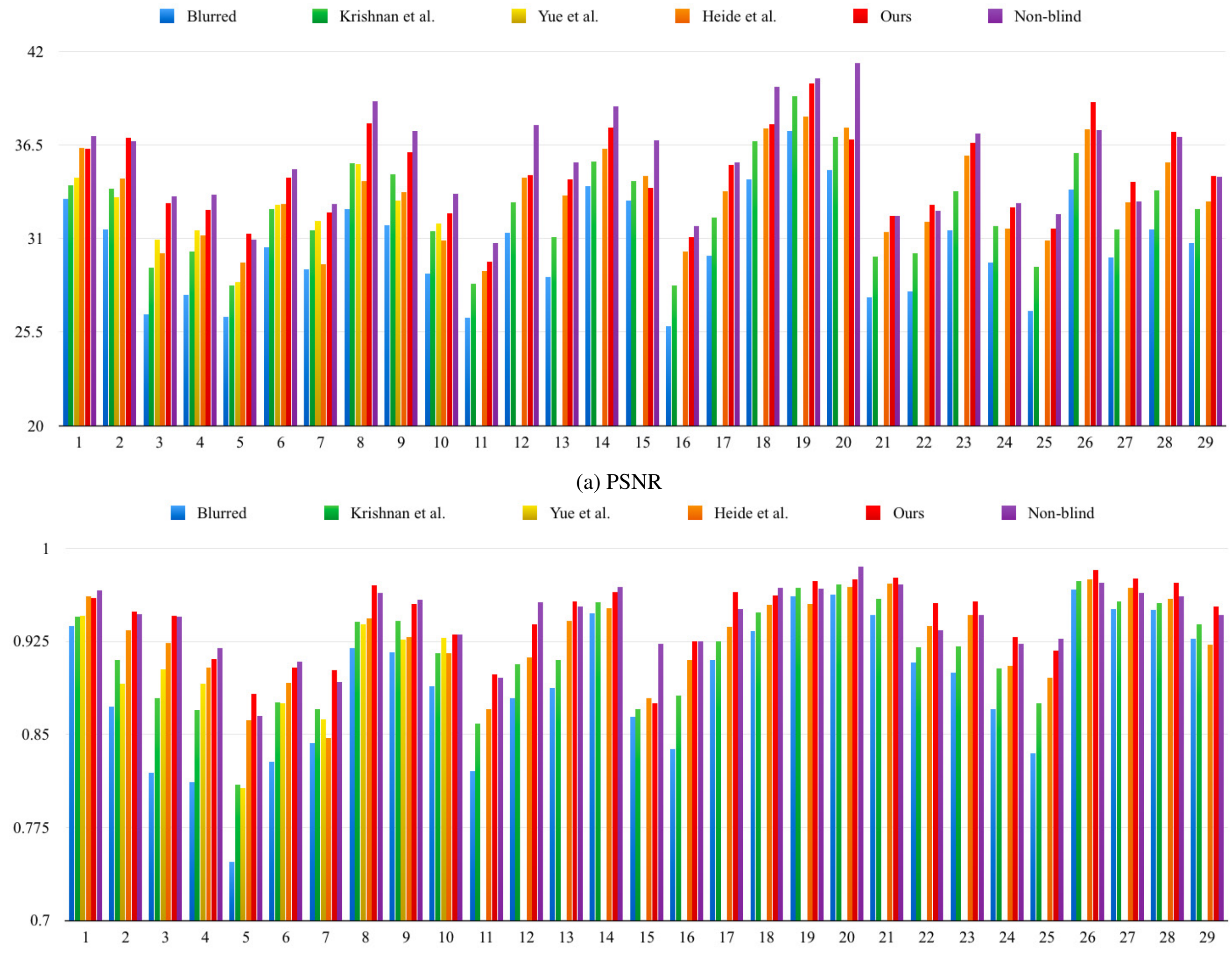
## 3. Additional synthesized results

In the main text, we have shown the averaged PSNR and SSIM assessments on 29 images selected from the dataset. Here we show the quantitative comparison between different algorithms on each image in Fig. 4. Notice that for most of the images, our algorithm outperforms others with respect to PSNR and SSIM values, and exhibits the closest result to the non-blind deconvolution version. For the results from Yue *et al.* [6], since we didn't have the source code, only the first 10 images are available for the comparison.

The full version of Fig. 7 in the main text is presented here in Fig. 5. Particularly, compared to state-of-the-art algorithms, ours successfully eliminates the color fringes all across the distorted images to the lowest level.

### 3.1. Comments on special case

We have also tested our algorithm on the images without noticeable blur. Via directly using the same settings in previous experiments, our algorithm works quite well on unblurred images. That said, for some regions of an image that suffer from less blurry effect, there is no need to fine tune our algorithm to avoid causing artifacts as many deconvolution algorithms may need.



(a) PSNR

(b) SSIM

Figure 4: Quantitative comparison results between different algorithms on 29 testing images. For almost all the testing images, our results out-perform others' and are the closest to the results of non-blind algorithm.

# 4. Additional real world results

## 4.1. Diffractive lens data

Full-size results are presented in Fig. 6. As shown in the images, chromatic aberrations have better corrections in our resolved ones, including large chromatic aberrations around highlights (see second patch of first row, and third patch of second row), those induced by sharp color changes (see second path of second row, and first patch of third row), and those due to defocus (red part in the second path of third row). Also, all these images exhibit clearer information than other results, especially in the third patch of first row and the first patch of fourth row, where the decoration pattern on the glass and the text on the back can be recognized more easily.

## 4.2. Refractive lens data

Full-size results are presented in Fig. 7. Similar to the previous results of diffractive optics, the results on refractive lenses show that our algorithm can handle different kinds of chromatic aberrations properly without sacrificing fine-grain details (first patch of first row,third batch of second row). Surprisingly, the results of our blind algorithm are even clear than a non-blind one (see the number in the second patch of first row). The results of correcting motion blur (fourth row) have also shown the ability of our algorithm to deal with more general deblur problem. However, there also exists some color fidelity problem in our results (see third patch of first row, and third patch of third row), which we hope to fix it using learning method or adaptive window size in the future.

## References

[1] F. Heide, Q. Fu, Y. Peng, and W. Heidrich. Encoded diffractive optics for full-spectrum computational imaging. *Scientific Reports*, 6, 2016.

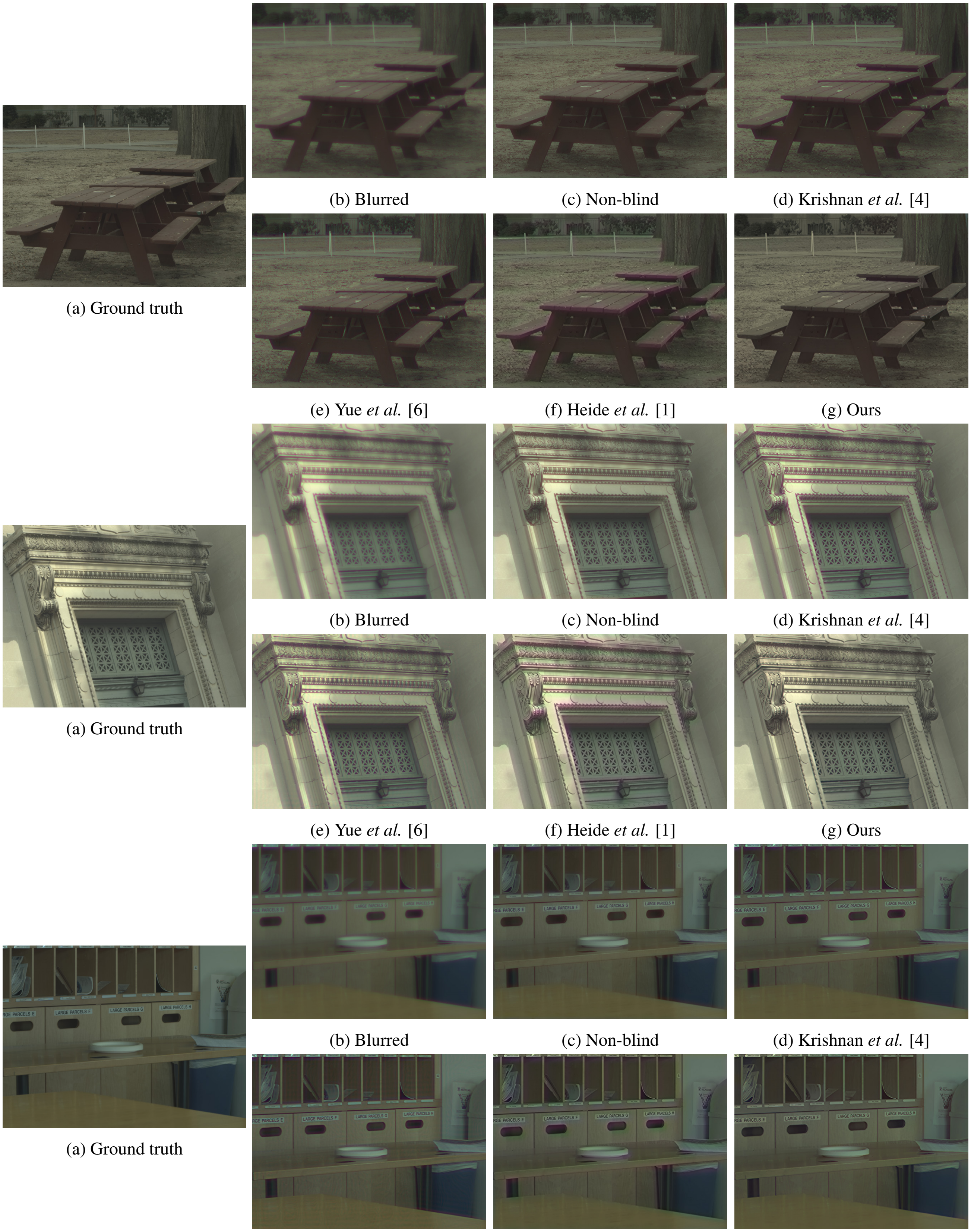
[2] F. Heide, M. Rouf, M. B. Hullin, B. Labitzke, W. Heidrich, and A. Kolb. High-quality computational imaging through simple lenses. *ACM Transactions on Graphics (TOG)*, 32(5):149, 2013.

[3] E. Kee, S. Paris, S. Chen, and J. Wang. Modeling and removing spatially-varying optical blur. In *Computational Photography (ICCP), 2011 IEEE International Conference on*, pages 1–8. IEEE, 2011.

[4] D. Krishnan, T. Tay, and R. Fergus. Blind deconvolution using a normalized sparsity measure. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 233–240. IEEE, 2011.

[5] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schölkopf. Non-stationary correction of optical aberrations. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 659–666. IEEE, 2011.

[6] T. Yue, J. Suo, J. Wang, X. Cao, and Q. Dai. Blind optical aberration correction by exploring geometric and visual priors. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.



(a) Ground truth

(b) Blurred

(c) Non-blind

(d) Krishnan *et al.* [4]

(e) Yue *et al.* [6]

(f) Heide *et al.* [1]

(g) Ours

(a) Ground truth

(b) Blurred

(c) Non-blind

(d) Krishnan *et al.* [4]

(e) Yue *et al.* [6]

(f) Heide *et al.* [1]

(g) Ours

(a) Ground truth

(b) Blurred

(c) Non-blind

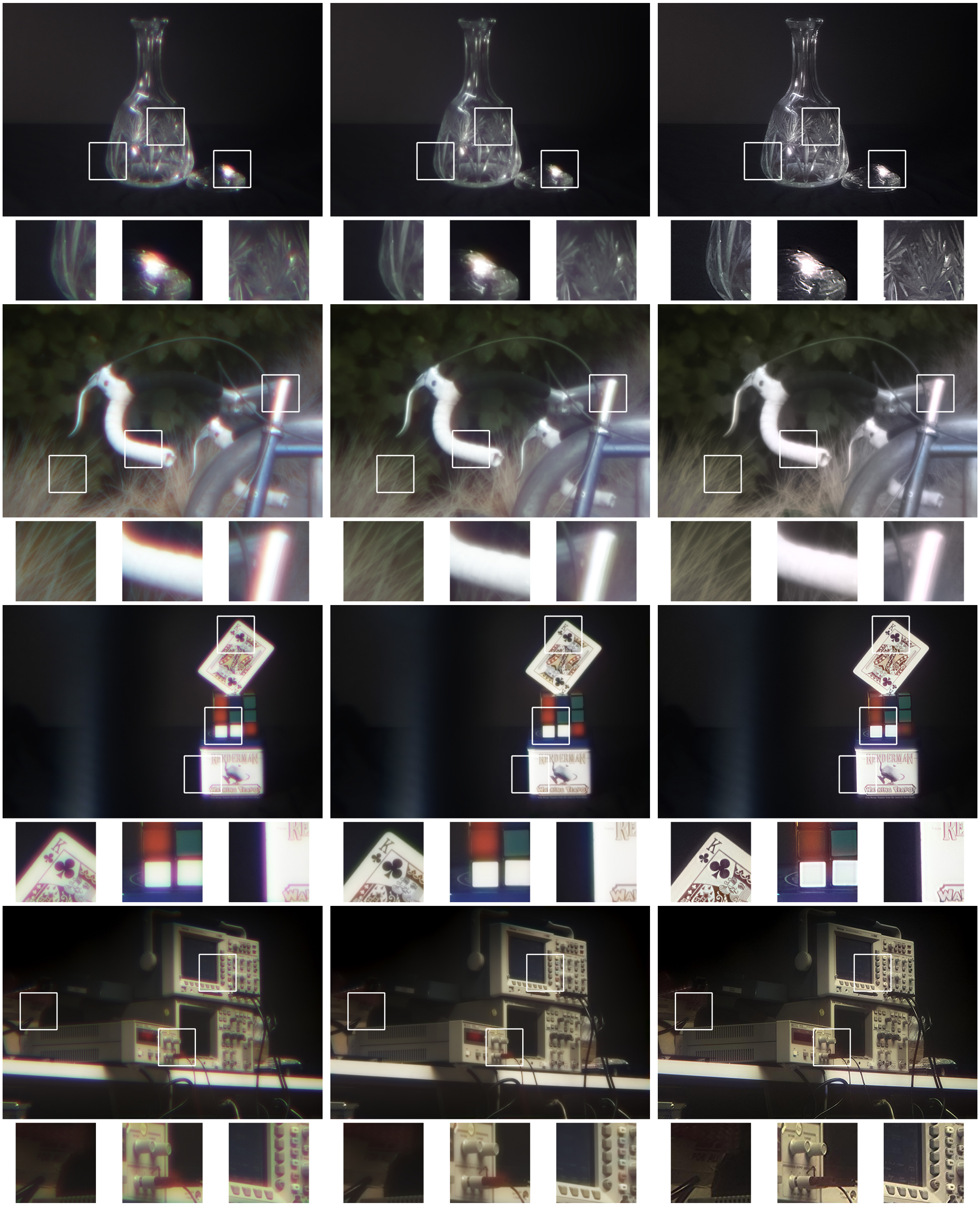
(d) Krishnan *et al.* [4]

(e) Yue *et al.* [6]

(f) Heide *et al.* [1]

(g) Ours

Figure 5: Full resolution version of images restored via different algorithms. Ours exhibit the least chromatic aberration artifacts compared with state-of-the-art works.

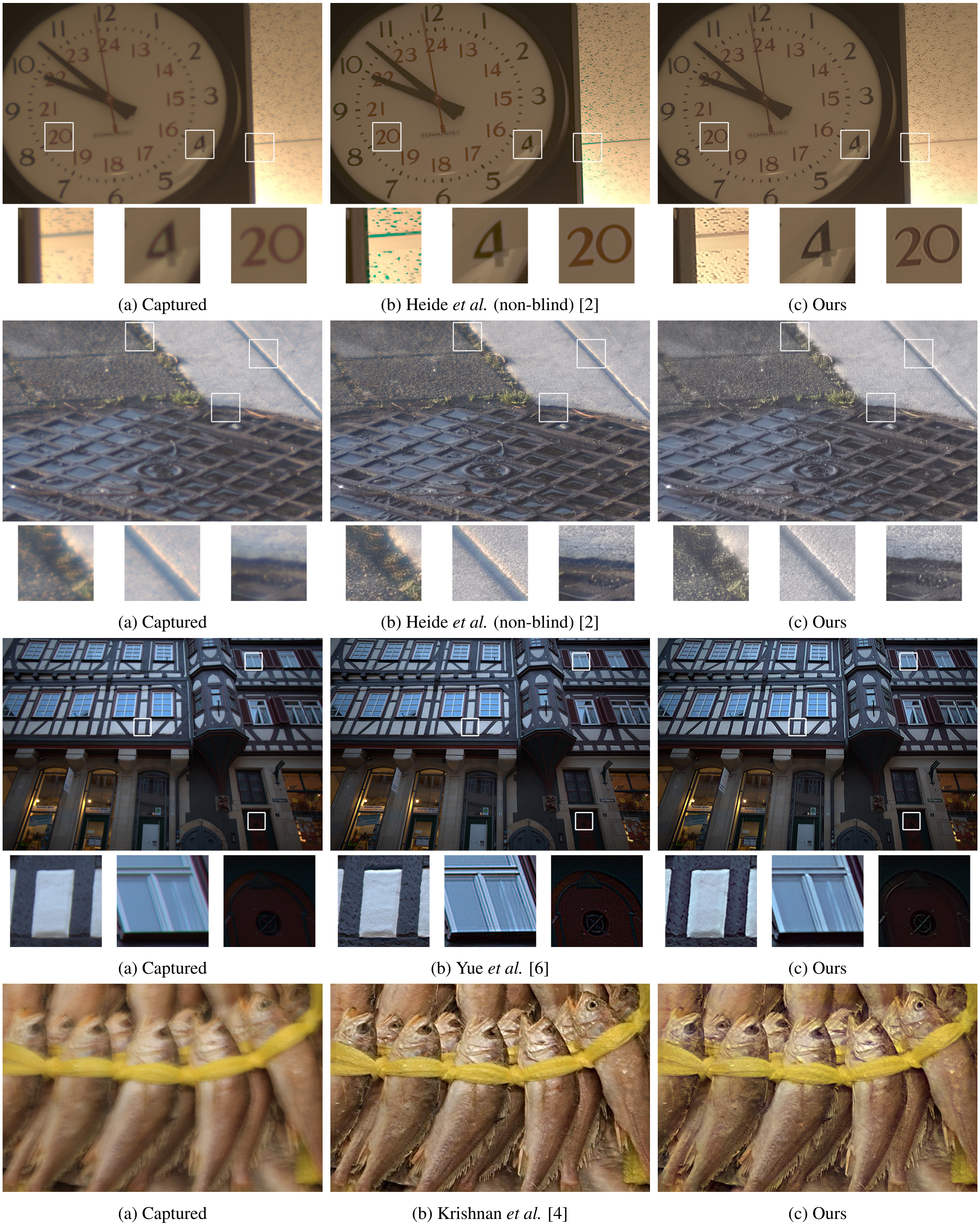


(a) Captured

(b) Heide *et al.*[1]

(c) Ours

Figure 6: Full-size restored images captured by a diffractive lens.



(a) Captured (b) Heide *et al.* (non-blind) [2] (c) Ours

(a) Captured (b) Heide *et al.* (non-blind) [2] (c) Ours

(a) Captured (b) Yue *et al.* [6] (c) Ours

(a) Captured (b) Krishnan *et al.* [4] (c) Ours

Figure 7: Full-size restored images captured by refractive lenses from state-of-the-art works.