

# Image quality enhancement using original lens via optical computing

Tao Yue<sup>1</sup>, Jinli Suo<sup>1</sup>, Yudong Xiao<sup>2</sup>, Lei Zhang<sup>2</sup> and Qionghai Dai<sup>1\*</sup>

<sup>1</sup>Department of Automation, Tsinghua University, Beijing, 100084, China

<sup>2</sup>Graduate School at Shenzhen, Tsinghua University, Shenzhen, 518055, China

\*[qionghaidai@tsinghua.edu.cn](mailto:qionghaidai@tsinghua.edu.cn)

**Abstract:** High-end lenses are usually composed of many optical elements to compensate various optical aberrations, e.g. geometric distortion, monochromatic and chromatic aberrations. The resulting complexity and machining accuracy requirements make high-end lenses too expensive, heavy, and fragile for day-to-day photography. To address this problem, we devised an optical computing approach to touch-up the low quality photos produced by simpler lenses. We propose a setup consisting of an easily accessible display and the original camera in order to perform optical aberration correction with a deconvolution framework. The equivalence of the degeneration model and the lens's optical computing turns the traditional blind deconvolution algorithm into its non-blind counterpart and promises robust performance. A prototype system is implemented to verify the feasibility of the proposed method, and a series of experiments on both synthetic and captured images are applied to demonstrate effectiveness and performance.

© 2014 Optical Society of America

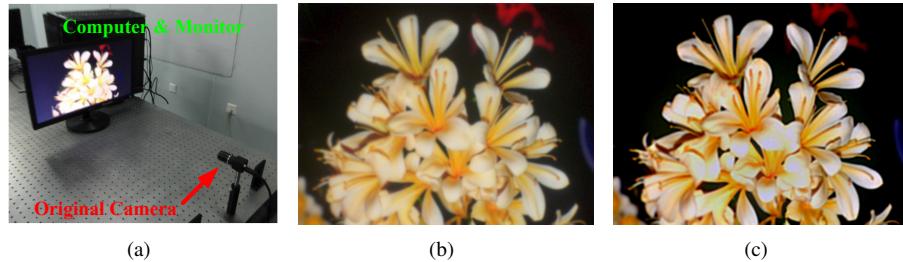
**OCIS codes:** (100.2980) Image enhancement; (100.1830) Deconvolution; (200.0200) Optics in computing.

---

## References and links

1. V. N. Mahajan and V. N. Mahajan, *Aberration theory made simple* (SPIE optical engineering press, Bellingham, Washington, USA, 1991).
2. E. Logean, E. Dalimier, and C. Dainty, "Measured double-pass intensity point-spread function after adaptive optics correction of ocular aberrations," *Opt. Express* **16**, 17348–17357 (2008).
3. H. Song, R. Fraanje, G. Schitter, H. Kroese, G. Vdovin, and M. Verhaegen, "Model-based aberration correction in a closed-loop wavefront-sensor-less adaptive optics system," *Opt. Express* **18**, 24070–24084 (2010).
4. L. Mugnier, J.-F. Sauvage, T. Fusco, A. Cornia, and S. Dandy, "On-line long-exposure phase diversity: a powerful tool for sensing quasi-static aberrations of extreme adaptive optics imaging systems," *Opt. Express* **16**, 18406–18416 (2008).
5. P. Bedggood, R. Ashman, G. Smith, and A. Metha, "Multiconjugate adaptive optics applied to an anatomically accurate human eye model," *Opt. Express* **14**, 8019–8030 (2006).
6. K. Wang, D. E. Milkie, A. Saxena, P. Engerer, T. Misgeld, M. E. Bronner, J. Mumm, and E. Betzig, "Rapid adaptive optical recovery of optimal resolution over large volumes," *Nature Methods* **11**, 625–628 (2014).
7. F. Heide, M. Rouf, M. B. Hullin, B. Labitzke, W. Heidrich, and A. Kolb, "High-quality computational imaging through simple lenses," *ACM T. Graphic* **32**, 149 (2013).
8. M. Hirsch, S. Sra, B. Scholkopf, and S. Harmeling, "Efficient filter flow for space-variant multiframe blind deconvolution," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, (IEEE, 2010), pp. 607–614.
9. E. Kee, S. Paris, S. Chen, and J. Wang, "Modeling and removing spatially-varying optical blur," in *Proceedings of IEEE International Conference on Computational Photography*, (IEEE, 2011).
10. T. Vettenburg and A. R. Harvey, "Holistic optical-digital hybrid-imaging design:wide-field reflective imaging," *Appl. Opt.* **52**, 3931–3936 (2013).

11. Q. Luo, L. Huang, N. Gu, and C. Rao, "Experimental study of a modified phase diversity with a diffraction grating," *Opt. Express* **20**, 12059–12066 (2012).
12. G. R. Brady, M. Guizar-Sicairos, and J. R. Fienup, "Optical wavefront measurement using phase retrieval with transverse translation diversity," *Opt. Express* **17**, 624–639 (2009).
13. N. Joshi, R. Szeliski, and D. J. Kriegman, "PSF estimation using sharp edge prediction," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, (IEEE, 2008).
14. Y. Shih, B. Guenter, and N. Joshi, "Image enhancement using calibrated lens simulations," in *European Conference on Computer Vision*, (Springer, 2012), pp. 42–56.
15. C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schölkopf, "Blind correction of optical aberrations," in *European Conference on Computer Vision*, (Springer, 2012), pp. 187–200.
16. S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, "An iterative regularization method for total variation-based image restoration," *Multiscale Modeling & Simulation* **4**, 460–489 (2005).
17. R. D. Nowak and M. A. Figueiredo, "Fast wavelet-based image deconvolution using the EM algorithm," in *Asilomar Conference on Signals, Systems, and Computers*, (2001), pp. 371–375.
18. M. Figueiredo and R. Nowak, "An EM algorithm for wavelet-based image restoration," *IEEE T. Image Process.* **12**, 906–916 (2003).
19. J. Starck, D. Donoho, and E. Candès, "Astronomical image representation by the curvelet transform," *Astronomy and Astrophysics* **398**, 785–800 (2003).
20. J. Starck, M. Nguyen, and F. Murtagh, "Wavelets and curvelets for image deconvolution: a combined approach," *Signal Processing* **83**, 2279–2283 (2003).
21. C. Vonesch and M. Unser, "A fast thresholded landweber algorithm for wavelet-regularized multidimensional deconvolution," *IEEE T. Image Process* **17**, 539–549 (2008).
22. D. Krishnan, T. Tay, and R. Fergus, "Blind deconvolution using a normalized sparsity measure," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, (IEEE, 2011).
23. A. Levin, R. Fergus, F. Durand, and W. T. Freeman, "Image and depth from a conventional camera with a coded aperture," *ACM T. Graphic* **26**, 70:1–70:10 (2007).
24. R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, "Removing camera shake from a single photograph," *ACM T. Graphic* **25**, 787–794 (2006).
25. Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," *ACM T. Graphic* **27**, 73:1–73:10 (2008).
26. N. Joshi, C. L. Zitnick, R. Szeliski, and D. J. Kriegman, "Image deblurring and denoising using color priors," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, (IEEE, 2009).
27. D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *Proceedings of IEEE International Conference on Computer Vision*, (IEEE, 2011).
28. R. A. Athale and W. C. Collins, "Optical matrix-matrix multiplier based on outer product decomposition," *Appl. Opt.* **21**, 2089–2090 (1982).
29. A. Dias, "Incoherent optical matrix-matrix multiplier," NASA Langley Research Center Opt. Inform. Process. for Aerospace Appl. **1**, 71–83 (1981).
30. R. A. Heinz, J. O. Artman, and S. H. Lee, "Matrix multiplication by optical methods," *Appl. Opt.* **9**, 2161–2168 (1970).
31. P. Guilfoyle and R. Stone, "Digital optical computer II," *Proc. SPIE* **1563**, 214 (1991).
32. H. Huang, L. Liu, and Z. Wang, "Parallel multiple matrix multiplication using an orthogonal shadow-casting and imaging system," *Opt. Lett.* **15**, 1085–1087 (1990).
33. A. Lewis, Y. Albeck, Z. Lange, J. Benchowski, and G. Weizman, "Optical computation with negative light intensity with a plastic bacteriorhodopsin film," *Science* **275**, 1462–1464 (1997).
34. M. O'Toole and K. Kutulakos, "Optical computing for fast light transport analysis," *ACM T. Graphic* **29**, 164:1–164:12 (2010).
35. D. Lefebvre, H. Arsenault, and S. Roy, "Nonlinear filter for pattern recognition invariant to illumination and to out-of-plane rotations," *Appl. Opt.* **42**, 4658–4662 (2003).
36. F. Yu, S. Jutamulia, T. Lin, and D. Gregory, "Adaptive real-time pattern recognition using a liquid crystal TV based joint transform correlator," *Appl. Opt.* **26**, 1370–1372 (1987).
37. P. Amba, "A short history of optical computing: rise, decline, and evolution," *Proc. SPIE* **7388**, 73880H (2009).
38. E. Leith, "The evolution of information optics," *IEEE J. Sel. Top. Quant.* **6**, 1297–1304 (2000).
39. C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schölkopf, "Non-stationary correction of optical aberrations," in *Proceedings of IEEE International Conference on Computer Vision*, (IEEE, 2011), pp. 659–666.
40. A. Kassir and T. Peynot, "Reliable automatic camera-laser calibration," in *Australasian Conference on Robotics and Automation*, (ARAA, 2010).
41. Z. Wang, A. C. Bovik, H. R. Sheikh, , and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE T. Image Process.* **13**, 600–612 (2004).



**Fig. 1. Prototype and results for image enhancement.** (a) The setup. (b) Low-quality input image. (c) Aberration-compensated result.

## 1. Introduction

Optical aberration is the deviation of real lenses with respect to the ideal imaging model, e.g. paraxial approximation, and can degrade image quality significantly. Although multiple lenses with different materials, curvatures, and interval distances are used to construct high-end aberration-corrected lens systems [1], optical aberrations in lens-based imaging systems are inevitable in practice. To obtain high-quality images, some researchers have proposed compensating for optical aberration by using adaptive optics during acquisition [2, 3, 4, 5]. Recently, Wang *et al.*[6] presented an adaptive optics method using direct wavefront sensing and descanned signal collection and measurement to manage spatially varying aberration of large volumes of data with a high refresh rate. Alternately, there is another type of method [7, 8, 9] where computational restoration is performed after data capture. Moreover, Vettenburg and Harvey [10] proposed a novel holistic design method to combine an optical acquisition system with post-processing in order to remove aberration effects in wide-field reflective cameras. This paper falls into the post-processing category, which requires less expertise and materials of wider availability.

Whether adopting adaptive-optics-based or post-processing methods, we need the characteristic data of the lens's aberration, i.e. the point spread functions (PSFs), beforehand. To derive the aberration information, researchers have proposed several kinds of methods, e.g. phase diversity retrieval [11, 12] and calibration [7, 9, 13, 14]. All these aberration calibration or estimation methods require a complex procedure or some special devices, such as checker board scheme, point light sources, etc. Without assistance from specific devices, Schuler *et al.*[15] assumed locally uniform aberrations and adopted blind deconvolution to estimate lens aberrations. This method eliminates the complex calibration procedure, but its robustness and performance can deteriorate in the case of serious aberrations. In sum, accurately obtaining the PSF of the lens is of vital importance but quite challenging.

Given the aberration data, one can use non-blind deconvolution to restore the latent image from a low-quality source. Firstly, the deconvolution needs to fit the convolution degeneration model iteratively to search for an optimal solution. Since lens aberration is often complex and spatially varying, the convolution must be conducted pixel-wise and is quite time consuming. Secondly, deconvolution is intrinsically ill-posed and regularizer terms defined from natural image statistics must be introduced to suppress unpleasant artifacts. Most of these methods impose sparsity constraints on the image gradients, e.g. total variation [16], wavelet coefficients [17, 18, 19, 20, 21], a hyper-Laplacian regularizer [22, 23], and other algebraic functions [24, 25]. Apart from spatial constraints, Joshi *et al.*[26] and Heide *et al.*[7] have also used color priority for regularization with impressive results. Alternately, Zoran and Weiss [27] use a patch-based learning model to regularize the restoration results. Still, the high complexity and

non-uniformity of lens PSFs make deconvolution extremely time-consuming.

Optical computing is an effective option for addressing the high complexity of computational lens-aberration compensation by making good use of the high speed and parallelism of light transport. Although its advantages in accelerating general computations [28, 29, 30, 31, 32, 33] (e.g., matrix multiplication, Fourier transformation, and matrix decomposition) have been overshadowed by the rapid advance of processor power, light-based computation is gaining momentum and has made considerable progress in certain areas. For example, O'Tool and Kutulakos [34] used a projector-camera system to compute light transport matrices, while Lefebvre *et al.*[35] and Yu *et al.*[36] used optical computing for pattern recognition. We refer readers to [37, 38] for a survey of this field.

With the aim of improving the image quality of low-end lenses simply and effectively, we propose a novel "touch-up" approach to address lens aberration via optical computing. Under the general framework of deconvolution, we use a display-camera system to perform calculations optically, as shown in Fig. 1(a). The convolution with ground truth PSF is conducted by capturing a displayed image using the original lens, and thus the deconvolution steps are implemented by performing a series of display–capture procedures. Since the priors expressed by a linear system can be easily incorporated into our optical computing framework, and without loss of generality, we implement the widely used sparse priors in the wavelet domain to validate our approach. Here the display–capture operation accurately simulates the degeneration model and eliminates the effects of PSF estimation error, so the proposed system achieves satisfactory performance, as shown in Fig. 1(b) and Fig. 1(c).

The proposed system is advantageous over previous methods in at least three ways. Firstly, the necessary devices are widely accessible and all the system calibrations and computing process are automatic, so our approach is easily implemented and offers a user-friendly interface. Secondly, using light transportation via the original lens to simulate the degeneration model, instead of convolving with the PSF-calibration or estimation results, is highly robust and shows promising performance. Thirdly, thanks to the efficiency of optical computation, we can restore a high-quality image in a matter of seconds. In addition, our approach provides a general optical computing framework for restoring a high-quality image from a low-quality input, and is applicable to arbitrary camera lenses.

Technically, the proposed method contributes mainly by the following three points:

- Formulation of the key modules in algorithms for lens aberration compensation into forward and backward convolution;
- Transformation of the backward convolution flexibly into a forward one, which can easily be implemented optically;
- Construction of a user-friendly prototype to verify the effectiveness of the proposed method.

## 2. Optical computing for aberration removal

Ignoring the out-of-focus effect, the projective model of a lens-based imaging system can be represented by a linear equation

$$\mathbf{B}(i, j) = \sum_{m,n} \mathbf{L}(m, n) \mathbf{K}(i, j, m, n) + \mathbf{N}(i, j), \quad (1)$$

where  $\mathbf{B}$  and  $\mathbf{L}$  are the observed image and the latent image, respectively,  $(i, j)$  and  $(m, n)$  are the indices of  $\mathbf{B}$  and  $\mathbf{L}$ , respectively,  $\mathbf{K}$  is the optical transfer function, and  $\mathbf{N}$  is additive noise that is assumed to be Gaussian. Ideally, the optical transfer function  $\mathbf{K}$  should be the unit impulse function, i.e., the imaging system accurately records the scene and the observed image

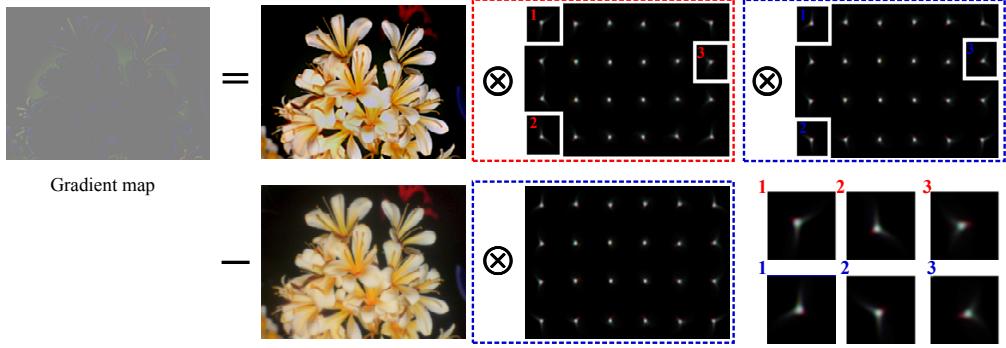


Fig. 2. **Diagram for calculating the gradient  $g_d$  in Eq. 7.** We use red and blue outlines to highlight two types of convolutions, i.e.  $\sum_{m,n} \mathbf{K}(i, j, m, n)(\cdot)$  and  $\sum_{i,j} \mathbf{K}(i, j, m, n)(\cdot)$ , respectively.

**B** exactly equals **L**. However, in real cases, because of optical aberrations, the optical transfer function becomes a low-pass filter and it projects a single pixel in **L** onto a set of pixels in **B**. Given a specific position  $(m_0, n_0)$  in **L**, its projection in **B**, i.e.  $\mathbf{K}(x, y, m_0, n_0)$ , is also known as the PSF or blur kernel of  $\mathbf{L}(m_0, n_0)$ . Usually the blur kernel caused by lens aberration is position dependent (i.e., spatially varying).

Assuming the observed image **B** and optical transfer function **K** are both known, the latent image can be derived by minimizing the following data fidelity term

$$E_d = \sum_{i,j} \left( \sum_{m,n} \mathbf{L}(m, n) \mathbf{K}(i, j, m, n) - \mathbf{B}(i, j) \right)^2, \quad (2)$$

which forces the synthetic record from the reconstructed latent image and optical transfer function to remain similar to the true image **B**. To suppress artifacts in the reconstructed image caused by lost high frequencies, many algorithms [17, 18, 19, 20, 21] based on wavelet domain sparse priors have been proposed. In this paper, the widely used wavelet term is applied for regularization because of its simplicity and effectiveness,

$$E_w(\mathbf{L}) = \|\mathcal{W}(\mathbf{L})\|_1, \quad (3)$$

with  $\mathcal{W}(\cdot)$  being the wavelet decomposition function, and the 'Symlets 8' wavelet is used here. Combining Eqs. 2 and 3, we get the final cost function

$$E = E_d + \lambda E_w, \quad (4)$$

with the coefficient  $\lambda$  balancing constraints from the fidelity term  $E_d$  and the regularization term  $E_w$ .

To minimize Eq. 4, we can apply the fast iterative shrinkage/threshold (IST) algorithm [17], which iteratively performs the following two steps after initialization  $\mathbf{L}^0 = \mathbf{B}$ :

(i) Updates the image by descending gradient

$$\mathbf{L}_g^t(i, j) = \mathbf{L}^t(i, j) - \tau \mathbf{g}_d(i, j); \quad (5)$$

(ii) Denoises the recovered image in the wavelet domain

$$\mathbf{L}^{t+1}(i, j) = \mathcal{T}_{\lambda \tau/2}(\mathbf{L}_g^t(i, j)). \quad (6)$$

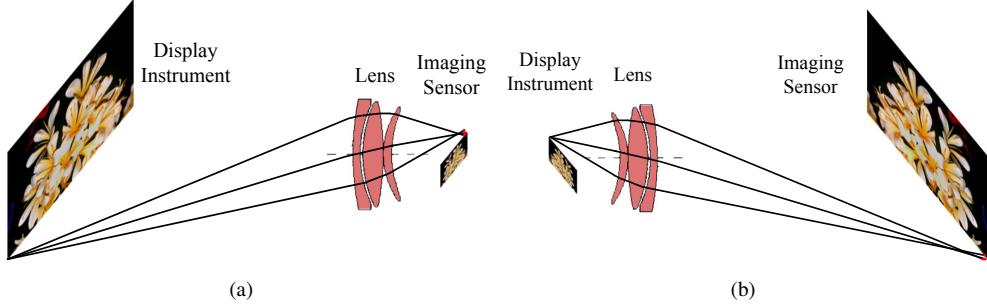


Fig. 3. **Optical computing for convolution modules.** (a) Forward convolution— $\sum_{m,n} \mathbf{K}(i, j, m, n)(\cdot)$ . (b) Backward convolution— $\sum_{i,j} \mathbf{K}(i, j, m, n)(\cdot)$ .

Here  $t$  is the iteration index,  $\tau$  is the updating step size,  $\mathbf{g}_d$  is the gradient of Eq. 2, and  $\mathcal{T}_{\lambda\tau/2}(\cdot)$  is the thresholding operation with  $\lambda\tau/2$  being the truncation threshold. In this paper,  $\tau$  is set to 0.8;  $\lambda$  needs to be tuned according to the noise level and will be discussed with the quantitative experiment. The gradient  $\mathbf{g}_d$  is calculated by

$$\mathbf{g}_d(m, n) = 2 \sum_{i,j} \mathbf{K}(i, j, m, n) \sum_{m,n} \mathbf{K}(i, j, m, n) \mathbf{L}(m, n) - 2 \sum_{i,j} \mathbf{K}(i, j, m, n) \mathbf{B}(i, j). \quad (7)$$

One can see that the calculation includes two main convolution modules: summation over indices  $(m, n)$  and  $(i, j)$  of the transfer function  $\mathbf{K}(i, j, m, n)$ . For clarity, we illustrate the gradient calculation in Fig. 2, with the two modules highlighted by red and blue rectangles.

In practice, it is often difficult to compute the gradient in Eq. 7 because it requires a highly accurate estimation of  $\mathbf{K}(i, j, m, n)$ , i.e., local PSFs at different image locations. Estimating these PSFs either blindly or using a checkerboard remains a challenging problem, although several works [8, 9, 14, 39] have proposed methods of simplification. In this paper, we try to solve this problem by replacing all the required PSFs with optical computing. Physically,  $\mathbf{K}$  is the transfer function of the original lens system, which moves the photons from  $(i, j)$  in the object plane to  $(m, n)$  in imaging plane. Thus, the summation operation  $\sum_{m,n} \mathbf{K}(i, j, m, n)(\cdot)$  in Eq. 7 exactly depicts the imaging process and can be computed by capturing an image using the original lens system, as shown in Fig. 3(a). Similarly, the remaining two summation operations  $\sum_{i,j} \mathbf{K}(i, j, m, n)(\cdot)$  can be computed by using a lens system and the transfer function  $\mathbf{K}'(m, n, i, j) = \mathbf{K}(i, j, m, n)$ , i.e., the original lens system after exchanging its object and imaging plane, as shown in Fig. 3(b). To simplify representation, in the following discussion we denote these two optical computing techniques as forward and backward convolution.

### 2.1. Optical computing for forward convolution

The forward convolution ( $FC(\cdot)$ ) describes the projective imaging from the latent sharp image  $\mathbf{L}$  to its aberration-corrupted version,

$$FC(\mathbf{L})(i, j) = \sum_{m,n} \mathbf{L}(m, n) \mathbf{K}(i, j, m, n). \quad (8)$$

Usually, the PSFs of the camera lens  $\mathbf{K}(i, j, m, n)$  vary at different  $(i, j)$ s and are very hard to estimate. In spite of recent progress in the calibration of camera PSFs[7, 9, 13], existing lens-aberration removal algorithms still suffer from the inaccuracy and limited range of PSFs estimation.

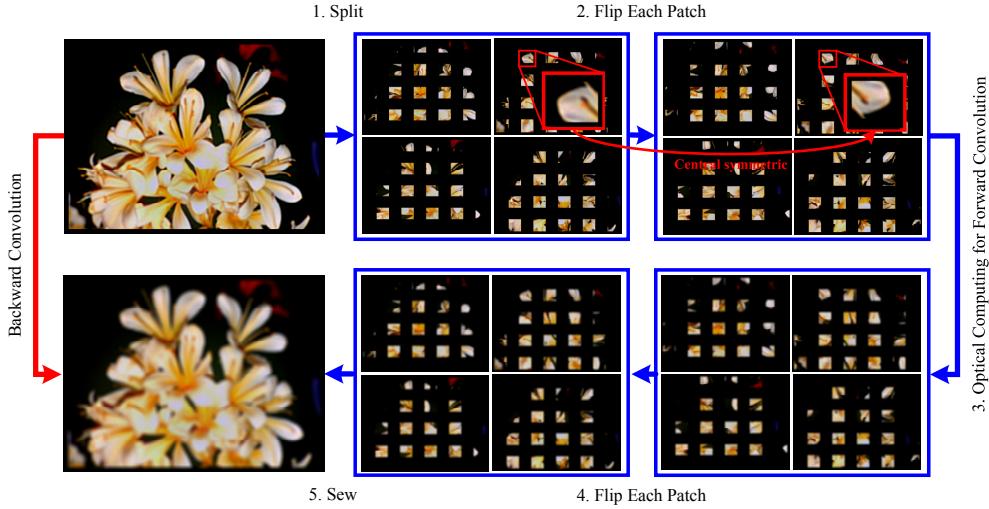


Fig. 4. Steps for approximating backward convolution using forward operation.

Let us return to the widely used deblurring framework for aberration compensation. The purpose of estimating a PSF is to perform a forward convolution corresponding to the degeneration process. Suppose the lens used to capture the low quality input is available, we can completely replace the estimate PSF convolution by capturing an image with the original camera. In practice, we can simply conduct the forward convolution by simulating the projection process with a display–camera system.

As shown in Fig. 3(a), the forward convolution of Eq. 8 in deblurring algorithms can be computed by imaging  $\mathbf{L}$  using the original lens and the result of forward operation  $\sum_{m,n} \mathbf{L}(m,n) \mathbf{K}(i,j,m,n)$  is accurately recorded on the image sensor. Therefore, we propose to design an optical computing system composed of the original camera and a display instrument to compute the forward convolution exactly and physically.

## 2.2. Optical computing for backward convolution

As discussed earlier, the backward convolution can be optically modeled by an ‘inverse’ optical lens, which means exchanging the object and imaging plane in the optical system computing forward convolution, as shown in Fig. 3(b). However, the system in Fig. 3(b) is not easy to build, because the required sensor size is significantly larger than that of a common camera CCD. Fortunately, the local PSFs caused by optical aberrations vary smoothly in the spatial domain, and thus can be regarded patch-wise as uniform [15, 39]. Based on this property, in this paper, a patch-based method is proposed to approximate the backward convolution using the original lens system.

Mathematically, the intensity at a given pixel  $(m,n)$  in  $\mathbf{L}$ ’s backward convolution  $BC(\mathbf{L})$  can be computed by the offset formula with respect to a certain pixel  $(x_0, y_0)$  within its neighboring area,

$$BC(\mathbf{L})(m,n) = \sum_{i,j} \mathbf{L}(i,j) \mathbf{K}(i,j,m,n) = \sum_{\Delta i, \Delta j} \mathbf{K}(x_0 + \Delta i, y_0 + \Delta j, x_0 + \Delta m, y_0 + \Delta n) \mathbf{B}(x_0 + \Delta i, y_0 + \Delta j), \quad (9)$$

where  $\Delta i = i - x_0$ ,  $\Delta j = j - y_0$ ,  $\Delta m = m - x_0$ , and  $\Delta n = n - y_0$  are the offsets.

Since the PSF  $\mathbf{K}(i,j,m,n)$  is approximately uniform in a local patch, for sufficiently small

$|\Delta x|$  and  $|\Delta y|$  we have

$$\mathbf{K}(i, j, m, n) \approx \mathbf{K}(i + \Delta x, j + \Delta y, m + \Delta x, n + \Delta y). \quad (10)$$

Let  $\Delta x = -\Delta m - \Delta i$  and  $\Delta y = -\Delta n - \Delta j$ , substituting Eq. 10 into Eq. 9 yields

$$BC(\mathbf{B})(x_0 + \Delta m, y_0 + \Delta n) \approx \sum_{\Delta i, \Delta j} \mathbf{K}(x_0 - \Delta m, y_0 - \Delta n, x_0 - \Delta i, y_0 - \Delta j) \mathbf{B}(x_0 + \Delta i, y_0 + \Delta j). \quad (11)$$

Defining  $\mathbf{B}'$  as the centrosymmetric (i.e. horizontally and vertically flipped) version with respect to  $(x_0, y_0)$  of the input image  $\mathbf{B}$ , we have  $\mathbf{B}'(x_0 - \Delta i, y_0 - \Delta j) = \mathbf{B}(x_0 + \Delta i, y_0 + \Delta j)$  and Eq. 11 becomes

$$BC(\mathbf{B})(x_0 + \Delta m, y_0 + \Delta n) \approx FC(\mathbf{B}')(x_0 - \Delta m, y_0 - \Delta n), \quad (12)$$

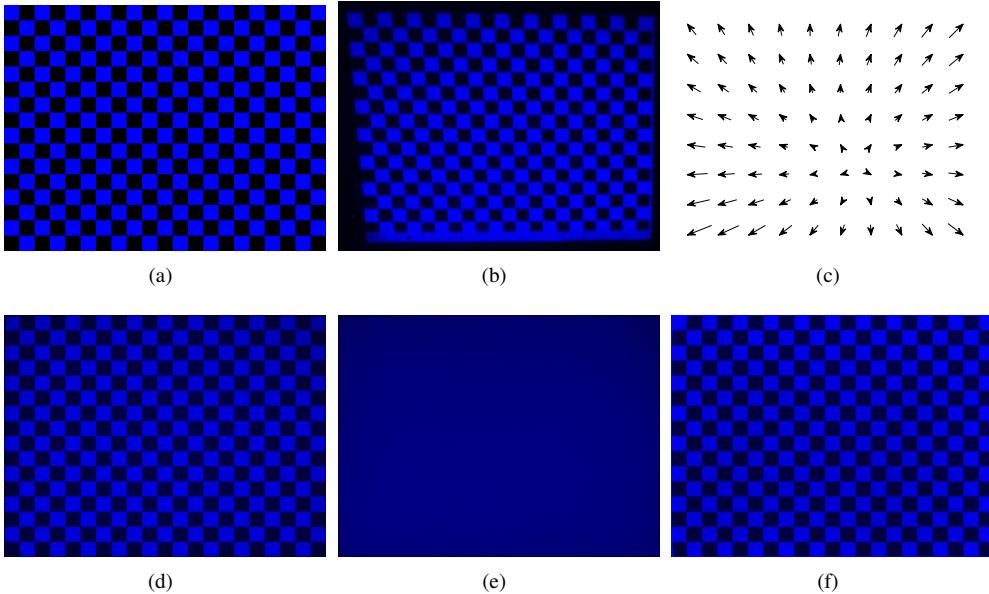
which is the centrosymmetric version of  $BC(\mathbf{B})$  with respect to  $(x_0, y_0)$  and can be computed easily by the original lens system. Since the offsets  $\Delta x = -\Delta m - \Delta i$  and  $\Delta y = -\Delta n - \Delta j$  need to be small to ensure high approximation accuracy, we split the images into sufficiently small patches and approximate their backward convolutions using the original lens system. In this paper, we refer to this approach as the *flip–FC–flip approximation*.

In our experiments, we computed the backward convolution of several patches simultaneously (in one snapshot) for an accelerated response. Because the flip manipulation would change the connection relationship along the division boundary (the pixels on the two sides of the border becomes nonadjacent after flipping), if we apply convolution to these adjacent patches simultaneously, the pixels on one side of the border will affect the result on the other side. To prevent this overlapping between adjacent patches, we split the image  $\mathbf{B}$  into a set of local patches arranged into four images to ensure that all the neighbors of a single patch do not appear in the same image. As illustrated in Fig. 4, the approximate version of  $BC(\mathbf{B})$  can be computed by the following steps: (1) Split the input image into four images to make sure the adjacent patches do not appear in the same image, as shown in Fig. 4; (2) Flip all the patches in both horizontal and vertical directions; (3) Display the four images and capture them using the original camera lens (forward convolution); (4) Flip the local patches (including the extended margin caused by aberrations) in the captured image in both horizontal and vertical directions; (5) Sew the four images together to form the final result.

### 3. Implementation details

Since the optical computing framework matches the physical imaging process quite well, the setup can be built easily by untrained users. The system includes a computer with display instrument (either monitor or projector) and the original camera that captures the input image. Specifically, the framework incorporates the forward and backward modules into a general deblurring framework and conducts several display-imaging processes to get the compensated results. The user only need to fix the camera and display instrument to ensure the display screen visible to the camera and the algorithm will calibrate the system (both the display instrument and the camera) and compute the final result automatically. Figure 1 shows our prototype system and the result for restoring a low quality image. In the following subsections, the system calibration will be detailed to validate the feasibility of the proposed approach.

**Geometric calibration.** Geometric calibration is used to build correspondence between the display and the image sensor. Instead of establishing and solving the projective geometry based model, we proposed to use the checkerboard pattern to compute the correspondence of the corner points and compute that of the rest pixels by spline interpolation. Considering the chromatic aberration, we perform geometric calibration in three channels separately.



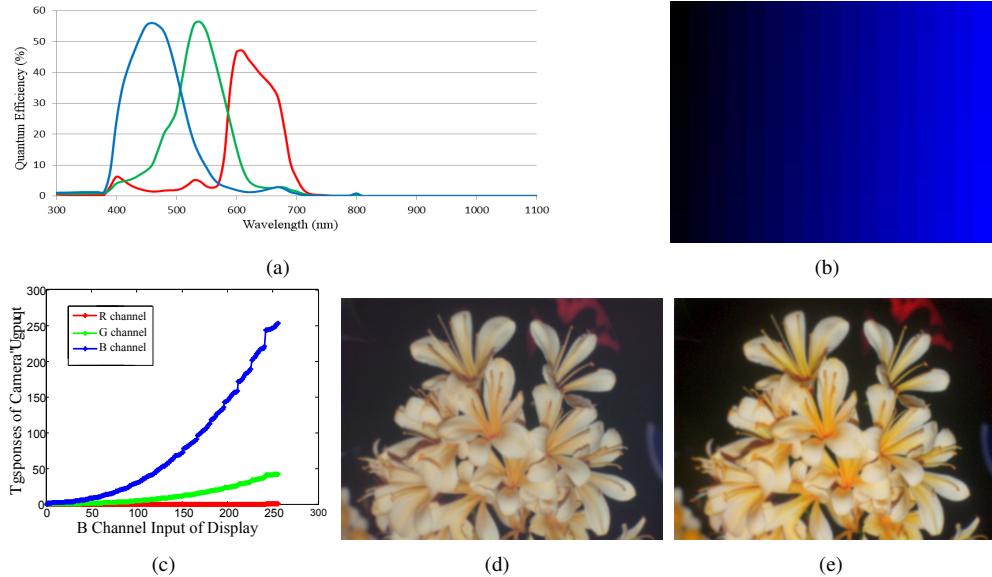
**Fig. 5. Geometric (a–d) and photometric calibration (e,f) in the blue channel.** (a) Original chessboard pattern. (b) Captured image. (c) Warping vectors of landmarks for interpolation. (d) Geometrically calibrated pattern. (e) Ratio image for dark-corner correction of the blue channel. (f) The chessboard pattern after correcting the geometry and dark corner distortions.

Figure 5 shows the adopted checkerboard pattern and the intermediate results of geometric calibration of the Blue channel. Firstly, the checkerboard pattern is displayed onto screen and captured with the camera, and then the corners are detected automatically by RADOCC algorithm [40] to derive the display-camera correspondence of these corners. Next, the rest correspondences are predicted by spline interpolation. After each display-capturing process, we warp the captured image (see Fig. 5(b)) according to the pre-calibrated correspondence before feeding it into the iterative framework, as shown in Fig. 5(d).

**Dark corner calibration.** To offset the vignetting effects in each channel, we displayed three images with constant intensity in the R, G, or B channel individually (intensity is set 128 with the dynamic range 0~255 to prevent underexposure or saturation), then computed the ratio images between the geometrically calibrated images and the input ones. Figure 5(e) shows the geometrically calibrated ratio image, which can be used to remove the dark-corner effect in the blue channel. Figure 5(f) shows the chessboard pattern following geometric and dark-corner correction.

**Cross-channel response calibration.** As is well-known, the three channels of a camera sensor do not respond to different wavelengths in a clear-cut manner. Figure 6(a) shows the response curves for the RGB channels of a camera (Point Grey FL3-U3-13S2C-CS), obviously there exists a large overlap between the B and G (also R and B) response curves. Since the colors of the both camera and the display device are generated by a composite of three basic colors determined by a color filter array, we propose to use a  $3 \times 3$  cross-channel response matrix to calibrate the camera responses with the color channels of the display device.

Considering that there exist both linear and non-linear color transformations during the



**Fig. 6. Cross-channel response calibration.** (a) Response curves of Point Grey FL3-U3-13S2C-CS. (b) Blue channel gradation map for calibrating the linear component  $\mathbf{C}$  and nonlinear component  $f_{r,g,b}$ . (c) Response curves of  $r$ ,  $g$ ,  $b$  channels vs. input channel  $b$ . (d) and (e) An image before and after cross-channel response calibration.

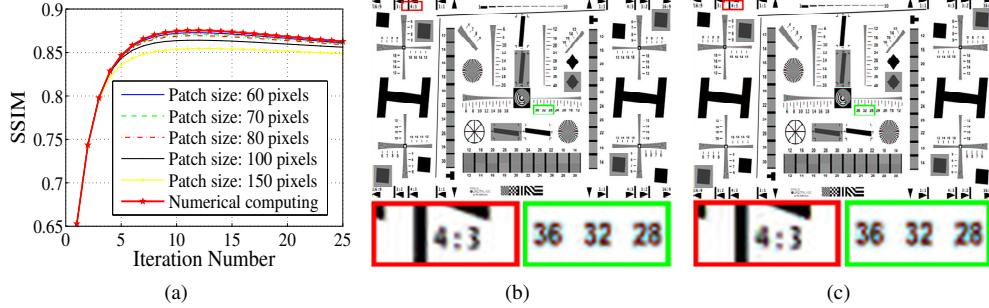
display–capture process, we formulate the cross-channel response of the camera sensor to the RGB display as

$$\begin{bmatrix} r' \\ g' \\ b' \end{bmatrix} = \begin{bmatrix} C_{r,r} & C_{g,r} & C_{b,r} \\ C_{r,g} & C_{g,g} & C_{b,g} \\ C_{r,b} & C_{g,b} & C_{b,b} \end{bmatrix} \begin{bmatrix} f_r(r) \\ f_g(g) \\ f_b(b) \end{bmatrix}. \quad (13)$$

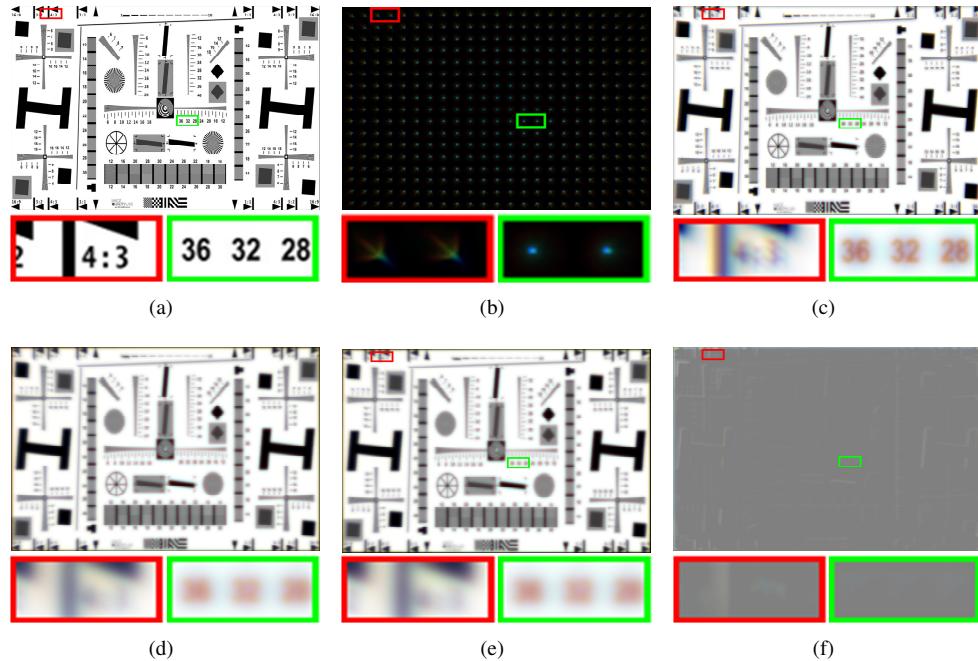
Here  $r$ ,  $g$ , and  $b$  are the three channel intensities of the displayed image and  $r'$ ,  $g'$ ,  $b'$  are those of the captured image,  $f_{r,g,b}(\cdot)$  is the nonlinear component, while the cross-channel matrix  $\mathbf{C} = \{C_{c_1,c_2}\}$  is the linear component with  $C_{c_1,c_2}$  describing the contribution of the display channel  $c_1$  to the camera channel  $c_2$ . In practice, we apply basic algebraic techniques to normalize the diagonal entries in  $\mathbf{C}$  to 1 for convenience.

To calibrate  $\mathbf{C}$  and  $f_{r,g,b}$ , we use intensity gradation maps in three channels. Taking the blue channel as an example, we first put the gradation map in Fig. 6(b) onto the display and perform geometric and dark-corner calibration on the captured image. Then, the camera response to each intensity level is computed by averaging over all the pixels at this level. Figure 6(c) shows the response curves of  $r'$ ,  $g'$ ,  $b'$  vs. the input  $b$  channel. The blue curve ( $b'$  vs.  $b$ ) is simply the nonlinear mapping  $f_b(\cdot)$ , and the scaling coefficient between blue curve and green/red curve is  $C_{b,g/r}$ .

Given the cross-channel matrix  $\mathbf{C}$  and the nonlinear response curve  $f_{r,g,b}$ , we can estimate the true intensity of an input image from its calibration. For each pixel in the observed image with intensities  $r'$ ,  $g'$ ,  $b'$ , the intensities of each corresponding input pixel can be derived by  $[r, g, b]^T = f_{r,g,b}^{-1}(\mathbf{C}^{-1}[r', g', b']^T)$ , which is implemented as a look-up table. Figures. 6(d) and 6(e) show the captured image before and after cross-channel response calibration.



**Fig. 7. The results of our approach with different patch sizes.** (a) The convergence curves of our method with backward convolution conducted by accurate numerical calculation and patch-wise flip–FC–flip approximations with different patch sizes. (b)(c) Restoration results by accurate numerical calculation and the proposed approximation with a 60-pixel wide patch size.

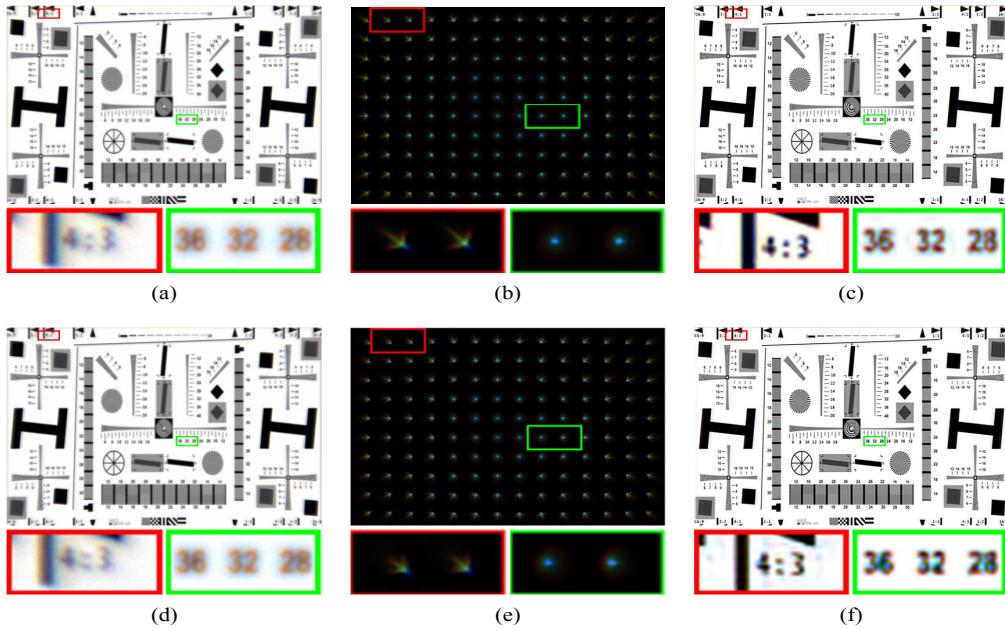


**Fig. 8. The error map of our flip–FC–flip approximation method.** (a) The ground-truth sharp image. (b) The local PSFs for the synthetic single lens camera. (c) Simulated image contaminated by optical aberrations (single lens). (d) Ground-truth image after backward convolution. (e) The approximate backward convolution result of our flip–FC–flip method. (f) The error map of our approximation.

## 4. Experimental results

### 4.1. Quantitative accuracy analysis of synthetic data

To verify the accuracy of our optical computing approach, a series of experiments and quantitative evaluations were conducted, with ground-truth results generated by simulating the imaging process of optical lenses using a ray tracing algorithm. In this experiment, we synthesized a single biconvex lens ( $f = 75\text{mm}$ ,  $f/2$ , bk7) imaging system with the display–camera distance set at 4 m and the Enhanced ISO12233 Resolution Chart at a resolution of  $1024 \times 768$  pixels



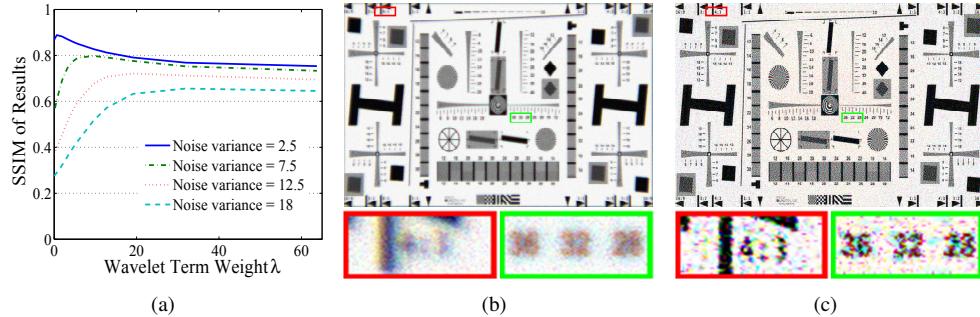
**Fig. 9. The results of our approach with different display–camera distances.** (a)(b) and (d)(e) show the captured degenerated images and corresponding PSFs with object distance at 40 cm and 4 m, respectively. (c)(f) display the results restoring the degenerated image in (a)(d) using the optical system with display–camera distance at 4 m and 40 cm, respectively.

used as a test image. The wavelet term weight  $\lambda$  was 3, and patch size was  $80 \times 80$  pixels. All the synthetic experiments were performed with these settings, except for the experiments specially designed for analyzing the influences of one of the above parameters.

**Approximate accuracy of the flip–FC–flip approach.** We first discuss the influence of patch size in the approximated backward-convolution operation. According to the mathematical analysis in Sec. 2.2, we know that the accuracy of our flip–FC–flip approximation method depends on the patch size setting. Here, we test the proposed method on different patch sizes and show the convergence performance of our deconvolution algorithm in Fig. 7(a). Considering the effectiveness of the structural similarity index (SSIM)(see [41] for details) in accessing the quality of blurred images, we adopt as our evaluation metric. We can see that although reduced patch size will yield better results, the influence of patch size becomes very small when smaller than  $80 \times 80$  pixels. Figure 7 shows the results from precise simulation of the forward and backward convolution operations in (b) and the result from applying our approximate method in (c) with 60-pixel wide patches. From Fig. 7(b) and 7(c), we see that our flip–FC–flip approximation approach can achieve almost the same result as that of the forward convolution. According to Fig. 7(a), we can see that the algorithm achieves optimal performance at around 10 iterations. Therefore, all the experimental results shown in this paper are derived with 11 iterations, except for the experiment comparing performance with varying iteration. (See Fig. 11.)

To analyze the accuracies at different image locations, we show the error map between the results of our flip–FC–flip approach and the exact numerical calculation. The PSF map is shown in Fig. 8(b). The ground-truth sharp image in Fig. 8(a) is placed at the objective plane, and the aberration-contaminated image is recorded on the image plane, as shown in Fig. 8(c). Figure 8(d) is the ground-truth result for the backward convolution of Fig. 8(a), by simulating the backward-projection system shown in Fig. 3(b), while Fig. 8(e) is the approximated result

using the flip–FC–flip method, with a patch size of  $80 \times 80$  pixels. It is obvious that our approximate approach can achieve very high accuracy, so that it is hard to find the difference between Fig. 8(d) and 8(e) visually. Therefore, for better comparison, Fig. 8(f) visualizes the absolute difference between our result and the ground truth. We can see that the approximate approach works fairly well in the center region, and the errors increase slightly in the corner regions where the local PSFs exhibit greater non-uniformity. However, these errors have negligible influence on the final restoration.



**Fig. 10. Restoration performance with different regularization weights and noise levels.** (a) SSIM curves with varying regularization weights and noise variances. (b) Aberration-corrupted images with noise variance at 18 (image intensity ranges from 0 to 255). (c) Our restoration result by setting the regularization weight to 18.

**Different display-camera distances.** The object distance of the camera capturing the input-degenerated image is usually different from the display–camera distance of our optical computing system. To test the influence of the different object distances of two systems on overall performance, we simulate a degenerated input image at given object–camera distance and conduct restoration using a setup with a different display–camera distance.

Figure 9(a) and 9(d) respectively show the degenerated images captured with 40 cm and 4 m object distance, and the corresponding ground-truth PSF maps are visualized in Fig. 9(b) and 9(e) respectively. We can see that the lens PSFs at different object distances as well as the corresponding degenerated images (Fig. 9(a) and 9(d)) are very similar. The deblurring results of the low quality images with 40 cm/4 m object distance using 4 m/40 cm display–camera distance system are shown in Fig. 9(c) and 9(f), from which we can see that although the two optical computing systems have different object distances from those during the original acquisition, the restoration results are still reasonable and promising. In other words, if the captured input image and our optical computing system are both well focused (no out-of-focus effect), we can neglect the difference in object distances during acquisition and restoration.

This phenomenon can be explained by the causation of optical aberration. It is known that most aberrations, including distortion, spherical aberration, coma aberration, chromatic aberration, etc., are caused by the outer regions of the camera lens and obvious with large aperture size or field of view. In other words, the aberration pattern of a specific camera is generally determined by its aperture size. Therefore, we need only to ensure consistency between the aperture sizes during acquisition and restoration of the input image. This also explains why the calibration-based methods [7, 14] need to calibrate the PSFs of a certain camera only once to restore all its captured images.

**Discussion on system noise and weight of regularization term.** The regularization term can help reduce the influence of system noise, so the weight  $\lambda$  needs to be adjusted according to the noise level. To this end, we conduct a synthetic experiment to analyze the relationship between noise level and the weight setting of the regularization term. We add Gaussian white noise after each simulated display-imaging operation, and test the performance of different regularization weights at different noise levels. For these experiments, we still use SSIM for quantitative evaluation, and the results are shown in Fig. 10(a). We can see that our algorithm achieves optimal performance when the regularization weight is near the noise variance. Figure 10(b) and 10(c) respectively show the simulated aberration-corrupted image and the final restoration result, with display-imaging noise superimposed (here noise variance = wavelet term weight  $\lambda = 18$ ). We can see that although the details are significantly enhanced, the final result still suffers from a high noise level. Therefore, we suggest a suppression of system noise by tuning the display instrument and system layout to make imaging noise as low as possible.



Fig. 11. Intermediate results of the first several iterations.

#### 4.2. Real cases

We built a prototype setup, shown in Fig. 1(a), to test the proposed approach on the low-quality images captured by a Point Grey camera (FL3-U3-13S2C-CS) and a single convex lens ( $f=6\text{mm}$ ,  $f/2$ ). An ASUS AS228 monitor with  $1920 \times 1080$  resolution was used as the display instrument, and the display–camera distance was around 40 cm. In our experiments, the patch size was set to  $80 \times 80$  pixels; wavelet term weight  $\lambda$  was 3. The intermediate results of our aberration compensation system are shown in Fig. 11. It is obvious that the proposed method converges to sharp resolution within a few iterations. Because of the elimination of the effects

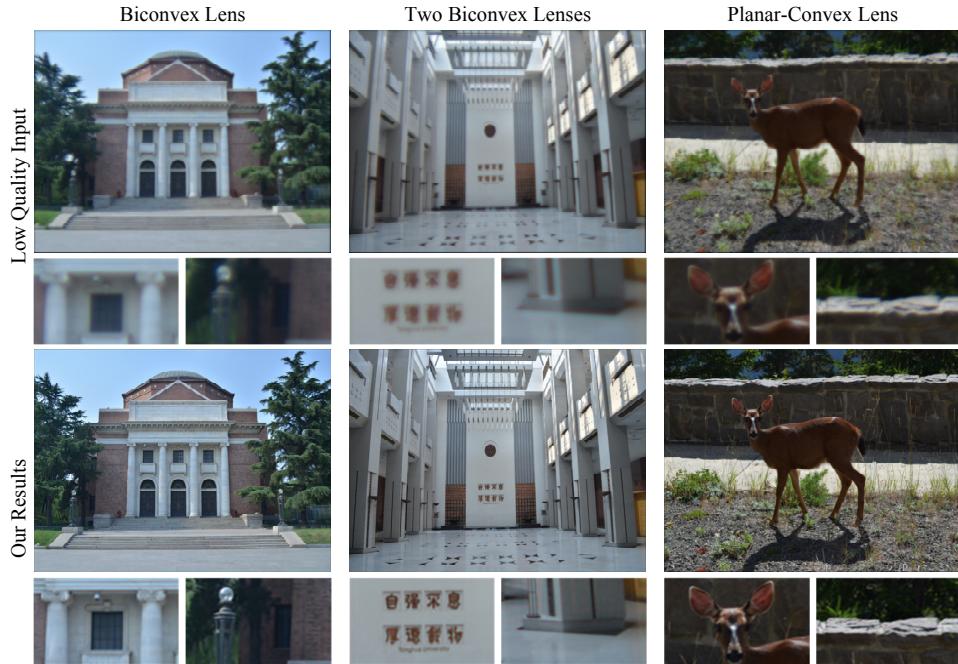


Fig. 12. More results for three different lenses.

from inaccurate PSFs, the compensation algorithm can converge robustly even without using any delicately designed deconvolution algorithm.

We also test the proposed approach on images captured by mounting various simple lenses, and results are shown in Fig. 12. These promising results for widely differing cases validate the applicability and high performance of our approach.

## 5. Summary and discussions

We proposed a framework for removing the optical aberration of low-end camera lenses via optical computing. This system, which is composed of a computer, a monitor, and the original camera, is easy to build and can be calibrated without any user interaction or the need of a calibration board. The proposed approach is advantageous in multiple aspects: high performance and robustness due to highly accurate forward and backward convolution, high efficiency provided by optical computation, and easy implementation with a user-friendly interface.

Although we limited our study to addressing depth-independent and smoothly varying degradations, the proposed framework is quite promising because these two assumptions hold for most lens aberrations. There are some other limitations and possible extensions for our approach: (1) The proposed system can only work in offline mode, which limits its applications in online image enhancement. (2) The algorithm assumes that the input image is completely focused and the display instrument is also placed within the depth of field of the camera to prevent the algorithm suffering from out-of-focus degradation. (3) The proposed method requires the optical computing system for processing every image, and cannot be calibrated beforehand.

Besides, it is worth noting that considering the limitation of the sensor resolution of the camera we used (Point Grey FL3-U3-13S2C-CS), the resolution  $1024 \times 768$  pixels is used in this paper. However, the proposed method can be applied for higher resolution if a high-end

camera sensor is available. The only limitation is the resolution of the monitor screen should be no smaller than that of the camera sensor.

In the future, the proposed method may be extended in the following respects: Firstly, the depth-independent limitation may be overcome with the progress of 3D-display instrumentation. Secondly, this work only tests a system with a simple deconvolution algorithm, and integrating more effective deconvolution into the framework is one of our future intentions and should improve the performance further. Also, building a parallel system that encapsulates the camera and the to-be-imaged object to extend the current system to online applications is a promising extension.