# REDUCING BOUNDARY ARTIFACTS IN IMAGE DECONVOLUTION

*Renting Liu* *Jiaya Jia*

The Chinese University of Hong Kong
Email: {rtliu, leojia}@cse.cuhk.edu.hk

## ABSTRACT

In image deconvolution, the *boundary value problem*, if not appropriately handled, often causes serious ringing artifacts in the restored results. This paper proposes a simple method to tackle this problem without any assumption on the noise level and the symmetry of the Point Spread Function (PSF). We establish new boundary conditions by smoothly expanding the input image to a large tile. It helps reducing the boundary discontinuities and accordingly makes all restoration filters based on Fast Fourier Transform (FFT) not produce obvious image border artifacts.

***Index Terms***— image deblurring, image deconvolution, ringing artifact, boundary condition

## 1. INTRODUCTION

Image blur is generally modeled as a convolution of a latent sharp image with a Point Spread Function (PSF) – that is, the blur filter. The convolution operator makes use of not only the image in the Field of View (FOV) of the given observation but also part of the scenery in the area bordering it. Hence, part of the information that is used to produce the filtered image boundary pixels is not available to the deconvolution process. Using the Fast Fourier Transform (FFT), the effect of missing boundary would propagate throughout the image and deteriorate the entire image. This problem is known as the *boundary value problem*, which poses a difficulty to various image restoration methods. Fig. 1(c) shows an example. When the missing pixel information immediately outside the border of an image is set as the mean value of the image, dominant horizontal and vertical oscillation patterns are yielded.

Depending on whether a spatial or Fourier domain restoration filter is used, there exist two solutions to the boundary value problem in practice. For a spatial domain filter, the missing pixel information outside the observed image can be synthesized by extrapolating the available image data. Various extrapolation methods have been proposed based on the selection of the Boundary Conditions (BC), including zero Dirichlet, periodic, reflective (also called Neumann or symmetric), and anti-reflective BC (see [1, 2] for the details). Some of these boundary conditions aim at removing the discontinuities at the boundary. For instance, reflective BC preserves the continuity of the image while anti-reflective BC retains the continuity of the image and its gradient. Most BC methods, such as those of [1, 2, 3], build large linear systems in the spatial domain in order to solve the deconvolution problem. They, thus, require much computational time and prefer strongly symmetric PSFs for effective matrix computation.

(a) (b) (c)

**Fig. 1**. Illustration of the boundary artifacts. (a) The latent clear image of size $256 \times 256$ with FOV shown in the white box. (b) Synthetic blurred image using a PSF of size $20 \times 20$ (shown at its bottom-right corner), where the missing boundary information is padded with the mean value of the whole image. (c) Restored image with severe boundary visual artifacts.

On the other hand, Fourier domain restoration filters, e.g., the Richardson-Lucy (RL) filter [4], adopt the Discrete Fourier Transforms (DFT) and assume the periodicity of the data. For a 2D DFT, this assumption implies that the left- and right-hand sides of the image are connected and so are the top and bottom boundaries. When DFT is performed, the missing pixels at the left-hand side will be taken from the right-hand side, and vice versa. Since the data obtained by DFT may not coincide with the missing ground truth data, discontinuities usually appear along the boundaries and the Gibbs oscillations (sometimes called *ripples*) are generated in the deconvolved image. Such oscillations can propagate inside the image and degrade the quality of the reconstruction prevailingly. A common way to remedy this problem is to extrapolate the image data at the border and meanwhile maintain the intensity or gradient smoothness. Linear interpolation between image boundaries or the 'edgetaper(·)' function in

The work was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. 412307) and is affiliated with the Microsoft–CUHK Joint Laboratory.

Matlab are widely used. Aiming at restoring astronomical photographs, the methods described in [5, 6] are based on the collected astronomical statistics and assume high levels of noise.

In this paper, we propose a simple method that combines the advantages of boundary conditions with the effectiveness of FFT to reduce the visual artifacts caused by the boundary value problem to a great extent. The main idea is to develop an interpolation method to smooth out the image boundaries using the tile generation. Our method does not require the symmetry of PSF and can be applicable to both natural images and astronomical photographs.

## 2. THE PROPOSED METHOD

We divide the image deconvolution into two stages. The first stage produces a *tile*, that is, an extrapolated image from the blurred observation. Next, a Fourier domain restoration filter is applied to the extrapolated image. Since any FFT-based restoration filter is applicable in our algorithm, we use the RL filter in all our experiments. Below we detail our image extrapolation scheme.

### 2.1. Tile Generation

We borrow the "tile" concept from the texture synthesis field and define it as a rectangular image block whose intensities at the left- and right-hand sides as well as the top and bottom boundaries follow some patterns. According to this definition, if a set of blocks are periodically tiled in a certain order, it can be guaranteed that no distinct discontinuities between neighboring blocks is observed.

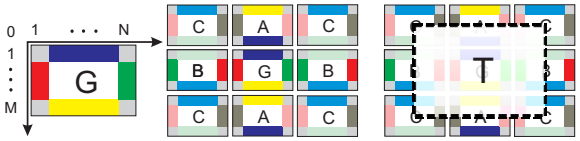



**Fig. 2**. Illustration of the tile generation. (a) A block (representing an input image) $G$ of size $M \times N$. (b) A set of tiles are arranged following an order such that the boundaries of adjacent blocks have the same intensity values (denoted by the color codes). (c) A rectangular tile $T$ is from the left-upper corner $(\frac{M+1}{2}, \frac{N+1}{2})$ to the right-bottom corner $(\frac{3M}{2}, \frac{3N}{2})$.

We show the creation of a valid tile $T$ in Fig. 2. The input image of size $M \times N$ is shown as a block $G$ in Fig. 2(a), each edge of which is assigned with a color. We design three padding blocks $A$, $B$, and $C$ with the same size as $G$, and arrange them in a manner shown in Fig. 2(b). This yields an expanded image of size $3M \times 3N$. These padding blocks satisfy the boundary constancy constraint, where the borders of adjacent blocks have the same intensity value (or color). It is illustrated using the paired colors in Fig. 2(b).

Let $(1, 1)$ denote the top-left pixel of the constructed tile. By cutting out the region from corner $(\frac{M+1}{2}, \frac{N+1}{2})$ to $(\frac{3M}{2}, \frac{3N}{2})$, we get an expanded image $T$, as shown in Fig. 2(c). $T$ has a desired characteristic for deconvolution using FFT, that is, it can be periodically laid out to form an infinitely large image without color discontinuities at the tile boundaries.

### 2.2. Padding Block Construction

With the above tile generation scheme, we introduce our method to construct the three blocks $A$, $B$, and $C$ such that the boundary problem can be reduced effectively. There are two main considerations. First, ringing artifacts are caused around sharp edges. We thus propose filling pixels with smooth colors between adjacent blocks. Second, when the FFT-based deconvolution methods are applied, in order not to cause the ringing artifact, the periodicity of the data should be guaranteed. The expanded image is with new borders that make it self-tileable.

Let $X(i,:)$ and $X(:,j)$, respectively, denote the $i$-th row $(i = 1, \ldots, M)$ and $j$-th column $(j = 1, \ldots, N)$ in image block $X$, where $X \in \{G, A, B, C\}$. In what follows, we first describe how $A$ is constructed. Blocks $B$ and $C$ are formed in a similar way and will be briefly discussed.

We expand block $A$ by a few pixels vertically, which results in a larger square block $A'$ containing $A$ and its outer-border $\partial A$. $\partial A$ is defined as

$$A'(i,:) = (H \otimes G)(M - \alpha + i,:), \quad (1)$$

$$A'(M + \alpha + i,:) = (H \otimes G)(i,:), \quad (2)$$

where $\alpha$ is the width of the new border $\partial A$ and $i \in \{1, \cdots, \alpha\}$. The new block $A'$ is of size $(M + 2\alpha) \times N$ and the border $\partial A$ contains the neighboring pixel information obtained from block $G$. $H$ is a Gaussian kernel to smooth out the image and reduce the noise. $\otimes$ is a convolution operator.

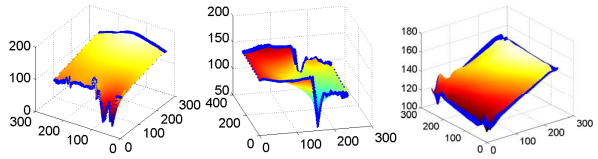
To achieve the second-order smoothness, we minimize the sum of second-order partial derivatives in block $A$. So we construct a membrane surface for the pixel values of $A$ defined as the solution of a minimization problem:

$$\min \iint_A |\Delta A|^2 \text{ with } A|_{\partial A} = A'|_{\partial A}, \quad (3)$$

where $\Delta . = \frac{\partial^2 .}{\partial x^2} + \frac{\partial^2 .}{\partial y^2}$ is the Laplacian operator. To make the solver efficient, we treat the boundary condition as a regularization term. The pixels in block $A$ are computed by solving a regularized least square problem

$$\min \left( \|\Delta A\|^2 + \lambda \|A - A'\|^2_{\partial A} \right), \quad (4)$$

where $\lambda$ is a weight that controls the pixel value smoothness of block $A$. In this way, the smooth transition within block $A$ and between $A$ and $G$ can be produced. Fig. 3(a) shows one



(a) (b) (c)

**Fig. 3**. The computed intensity values in blocks (a) $A$, (b) $B$, and (c) $C$ for the example in Fig. 1(b). Points used to set boundary conditions are shown in blue.

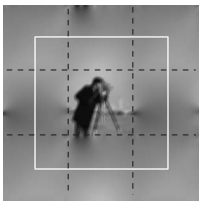


**Fig. 4**. Illustration of the tile generation. The rectangular region in the central white box indicates a valid *tile*.

example of the computed block $A$. Outer-border pixels are shown in blue.

Similarly, block $B$ is first expanded to $B'$ of size $M \times (N + 2\alpha)$ horizontally. So the new outer-border $\partial B$ is constructed as $B'(:, i) = (H \otimes G)(:, N - \alpha + i)$ and $B'(:, N + \alpha + i) = (H \otimes G)(:, i)$, where $i \in \{1, \cdots, \alpha\}$. Then block $B$ is computed in a way similar to estimating $A$.

After we compute blocks $A$ and $B$, block $C$ is symmetrically expanded into a block $C'$ of size $(M + 2\alpha) \times (N + 2\alpha)$ with a new border $\partial C$ defined as

$$C'(i,:) = B'(M - \alpha + i,:), C'(M + \alpha + i,:) = B'(i,:),$$
$$C'(:, i) = A'(:, N - \alpha + i), C'(:, N + \alpha + i) = A'(:, i),$$

where $i \in \{1, \cdots, \alpha\}$. Then block $C$ is computed similarly by solving a linear optimization problem with regularization terms. The final expanded image is shown in Fig. 4 with a central white box indicating a valid tile.

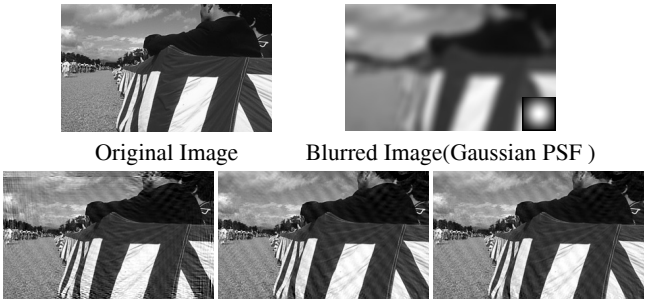
The restoration filter is applied to a tile of size $2M \times 2N$, 4 times of the original blurry image size. The RL deconvolution algorithm requires 4 FFTs per iteration and hence incurs a cost of $O(N^2 \log_2 N)$ operations for an $N \times N$ image. The computational cost on our expanded tile is increased by a factor of about 16 (when $M = N$). However, it can be reduced by using smaller-size blocks $A$, $B$, and $C$.

## 3. EXPERIMENTS

We conducted several experiments using symmetric and non-symmetric PSFs and compare them with reflective BC[1], anti-reflective BC[2], and the 'edgetaper(·)' function in Matlab. Besides visual evidences, we also provide numerical analysis, which calculates and compares Boundary Peak Signal-to-Noise Ratio (BPSNR) in the critical boundary regions of the reconstructed image. In our experiments, we regard the points, with their distance to the image border smaller than $0.1 * \min(M, N)$, as boundary points.

Different levels of noises are added to the blurred images to test the effectiveness of our boundary artifact reduction algorithm. In experiments, zero mean white Gaussian noises are added and the noise level in each example is measured by the Signal-to-Noise Ratio (SNR), given by $20 \log_{10}(\sigma_G / \sigma_n)$, where $\sigma_G$ and $\sigma_n$ are the standard deviations of the blurred image $G$ and noise, respectively. All our experimental results are acquired by using the RL algorithm with 200 iterations and setting $\alpha = 10$ and $\lambda = 1$.

*Result 1*: Comparison of the deconvolution results for an input image blurred with a symmetric PSF and no additive noise. As shown in Fig. 5, our algorithm generates high quality restoration results and comparable BPSNR to anti-reflective BC algorithm. We suggest readers to view these figures in their original resolution.



Original Image Blurred Image(Gaussian PSF )

(a) BPSNR = 40.8 (b) BPSNR = 52.7 (c) BPSNR = 54.5

**Fig. 5**. Restoration results with symmetric PSF and no additive noise. (a)-(c) Restoration results of reflective BC [1], anti-reflective BC [2], and our algorithm, respectively.

(a) BPSNR = 50.1 (b) BPSNR = 53.4 (c) BPSNR = 56.5

**Fig. 6**. Restoration results with a non-symmetric PSF (Fig. 1(b)) and additive noise (SNR = 92.8). (a)-(c) show restoration results using 'edgetaper(·)' in Matlab, anti-reflective BC, and our algorithm, respectively.

*Result 2*: Comparison of the deconvolution results for an image blurred with a non-symmetric PSF and additive noise. As shown in Fig. 6, both anti-reflective BC and our algorithm suppress the ripples near the image border and outperform 'edgetaper(·)'.

*Result 3*: Comparison of the BPSNR-SNR curves of dif-

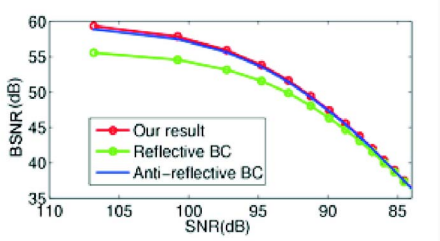



**Fig. 7**. The BPSNR-SNR curves of different algorithms.

(a) (b) (c)

**Fig. 8**. (a) Original Image. (b) Blurred image using a PSF of size $20 \times 20$ (shown at its bottom-right corner). (c)The restoration result of our algorithm (BPSNR = 46.4).

ferent algorithms using the example in Fig. 1. As illustrated in Fig. 7. our algorithm yields comparable restoration performance to anti-reflective BC and better performance than reflective BC.

*Result 4*: Comparison of the average computation time of different algorithms for the examples in Figs 5 and 6. Table 1 shows that our algorithm runs a bit slower than 'edgetaper(·)' which requires no extrapolation of the blurry image.

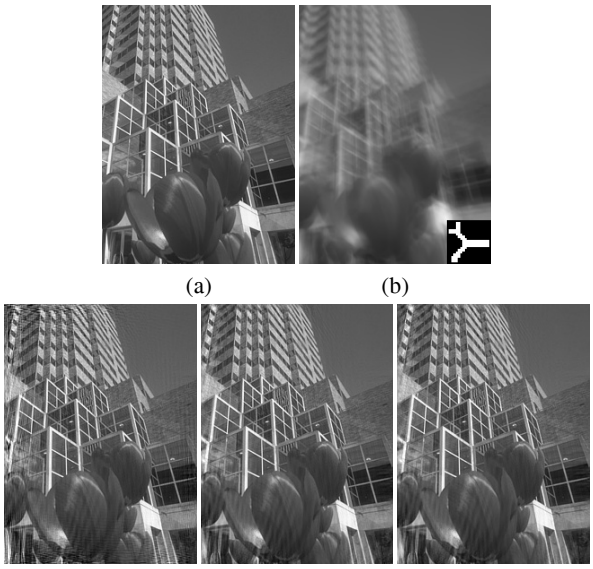
**Table 1**. Average running time of different algorithms.

| Algorithm | Test 1(s) | Test 2(s) |
|---|---|---|
| reflective | 643 | 135 |
| anti-reflective | 560 | 146 |
| edgetaper(·) | 39 | 40 |
| our algorithm | 68 | 62 |

More experimental results and comparisons are shown in Figs. 8 and 9.

## 4. CONCLUSIONS

In this paper, we have proposed a simple method for reducing the boundary ringing artifacts in image deconvolution. Our method combines the advantages of boundary smoothness with FFT to reduce the artifacts caused by the boundary value problem. It does not require the symmetry condition of the PSF and can be adopted by any FFT-based restoration methods. Experiments show that our algorithm yields quantitatively satisfying restoration results with reasonably short running time.



(a) (b)

(c)BPSNR = 37.6 (d)BPSNR = 39.9 (e) BPSNR = 40.0

**Fig. 9**. (a) Original Image. (b) Blurred image using a PSF of size $20 \times 20$ (shown at its bottom-right corner). (c)-(e) show the restoration results of reflective BC [1], anti-reflective BC [2], and our algorithm, respectively.

## 5. REFERENCES

[1] R. Chan M. Ng and W. Tang, "A Fast Algorithm for Deblurring Models with Neumann Boundary Conditions," *SIAM J. Sci. Comput.*, vol. 21, pp. 851–866, 1999.

[2] A. Martinelli M. Donatelli, C. Estatico and S. Serra-Capizzano, "Improved image deblurring with anti-reflective boundary conditions and re-blurring," *Inverse Problems*, vol. 22, no. 6, pp. 2035–2053, 2006.

[3] Daniela Calvetti and Erkki Somersalo, "Statistical elimination of boundary artifacts in image deblurring," *Inverse Problems*, vol. 21, pp. 1694–1714, 2005.

[4] L. Lucy, "Bayesian-based iterative method of image restoration," *Journal of Astronomy*, vol. 79, pp. 745–754, 1974.

[5] M. Bertero and P. Boccacci, "A simple method for the reduction of boundary effects in the Richardson-Lucy approach to image deconvolution," *Astronomy & Astrophysics*, vol. 437, pp. 369–374, 2005.

[6] M. Donatelli R. Vio, J. Bardsley and W. Wamsteker, "Dealing with edge effects in least-squares image deconvolution problems," *Astronomy & Astrophysics*, vol. 442, pp. 397–403, 2005.