

## RESEARCH

# Efficient Non-uniform Deblurring Based on Generalized Additive Convolution Model

Hong Deng, Dongwei Ren, Wangmeng Zuo\*, Hongzhi Zhang and Kuanquan Wang

## Abstract

Image with non-uniform blurring caused by camera shake can be modeled as a linear combination of the homographically transformed versions of the latent sharp image during exposure. Although such a geometrically motivated model can well approximate camera motion poses, deblurring methods in this line usually suffer from the problems of heavy computational demanding or extensive memory cost. In this paper, we develop generalized additive convolution (GAC) model to address these issues. In GAC model, a camera motion trajectory can be decomposed into a set of camera poses, i.e., in-plane translations (slice) or roll rotations (fiber), which can both be formulated as convolution operation. Moreover, we suggest a greedy algorithm to decompose a camera motion trajectory into a more compact set of slices and fibers, and together with the efficient convolution computation via fast Fourier transform (FFT), the proposed GAC models concurrently overcome the difficulties of computational cost and memory burden, leading to efficient GAC-based deblurring methods. Besides, by incorporating group sparsity of the pose weight matrix into slice GAC, the non-uniform deblurring method naturally approaches toward the uniform blind deconvolution. Experimental results show that GAC-based deblurring methods can obtain satisfactory deblurring results compared to both state-of-the-art uniform and non-uniform deblurring methods, and are much more efficient than non-uniform deblurring methods.

**Keywords:** camera shake; image deblurring; non-uniform deblurring; blind deconvolution; fast Fourier transform

## 1 Introduction

Image blur is generally inevitable due to various factors such as defocus, camera shake, etc. Blind deblurring from a real world blurry image that needs to estimate the blur procedure and latent sharp image is a very ill-posed problem. Recently, several hardware-assist approaches had been developed [1–6], by which additional information can be acquired to reduce the ill-posedness of the blind deblurring. These hardware-assist approaches can provide much easier deviation of the blur or sharp image, but require complex camera configurations [1, 2] or dedicatedly designed hardware support [3–6], far from taking the place of traditional imaging devices. At the same time, in the era of ubiquitous acquisition of digital images using portable imaging devices, e.g., digital camera, mobile phone, camera shake is often unavoidable during exposure procedure, a major cause that ruins a photograph. For effective and efficient deblurring, it is crucial to develop appropriate forward blur model that can well explain the

image degradation process of camera shake, and then to design proper regularizers and efficient optimization algorithms.

Earlier approaches on camera shaken deblurring usually assume that the blur is spatially invariant, due to which the blurry observation can be simply modeled as convolution of sharp image and blur kernel, and deblurring can thus be modeled as a blind deconvolution problem [7–12], where much attention had been devoted to design effective optimization algorithms. Since the convolution operation can be computed efficiently via fast Fourier transform (FFT), these blind deconvolution algorithms are commonly efficient. On one hand, Fergus et al. [7] adopted a mixture-of-Gaussians for representing the distribution of gradients, and introduced a variational Bayesian (VB) approach [13] to estimate the blur kernel. The theoretical and experimental analysis by Levin et al. [8] demonstrated the advantages of variational Bayesian (VB) over maximum a posterior (MAP), and inspired a number of VB-based blind deconvolution algorithms [7, 8, 14, 15]. On the other hand, by enforcing sparser priors [10, 16, 17]

\*Correspondence: cswmzuo@gmail.com

School of Computer Science and Technology, Harbin Institute of Technology, 92, West Dazhi Street, 150001, Harbin, China

or exploiting edge prediction step [9, 18–21] to select salient edges in latent image, several carefully designed MAP algorithms can also exhibit promising performance. Most recently, discriminative learning approaches had been developed to learn proper priors for better estimating blur kernel [22–25].

However, camera shake blur is often spatially varying, and consequently cannot be simply modeled as a single blur kernel. Thus, how to reasonably model camera shake blur plays a central role in non-uniform deblurring problem that can be categorized from 2 aspects. One direct strategy is to approximate the non-uniform blur as multiple blur kernels, where the blurry image is divided into several regions, and each region takes uniform blur. The multiple blur kernels can then be estimated from each region by performing uniform blind deconvolution method locally [26]. However, this approach does not consider the global constraints on local blur kernels based on camera motion trajectory. The other geometrical strategy is based on the projective motion path blur model (PMPB) [27], where a camera shake trajectory can be decomposed into a sequence of camera poses lying in the 6D camera pose space ( $x$ -,  $y$ -,  $z$ -axes rotation, and  $x$ -,  $y$ -,  $z$ -axes translation), and consequently the sharp image projected by each pose is a homography, which is then weighted according to its exposure time, resulting in the blurry image. By far, there are mainly two simplified 3D geometrical models to approximate the 6D subspace, i.e., Whyte et al. [28] suggested to employ  $x$ -(pitch),  $y$ -(yaw), and  $z$ -(roll) axes rotation, while Gupta et al. [29] proposed to adopt  $x$ - and  $y$ -axes (in-plane) translation and  $z$ -axis rotation. By combining the global camera motion constraint and the efficient filter flow (EFF) framework [30], Hirsch suggested to construct local uniform blur models guided by the geometrical constraint [31].

Even in the simplified 3D camera pose subspace [28, 29], non-uniform deblurring methods still suffer the problem of high computational cost or extensive memory burden, placing prominent restriction on its wide applications. To speed up the blur estimation step, Gupta et al. [29] pre-computed a sparse matrix for each homography transform, and thus the forward non-uniform blur operator can be equivalently defined as the weighted sum of the homography transform matrices, and Hu and Yang [32] further restricted the possible camera pose in a low dimensional subspace. Although the pre-computation techniques can relatively relax the computational inefficiency, these accelerated methods [29, 32] take increasing memory burden to store the huge homography transform matrices. Even so, subsequent computation of homography transform matrices is still computationally costly, and thus it

is interesting to ask that *is it possible to design forward blur model to benefit from FFT for efficient non-uniform deblurring?*

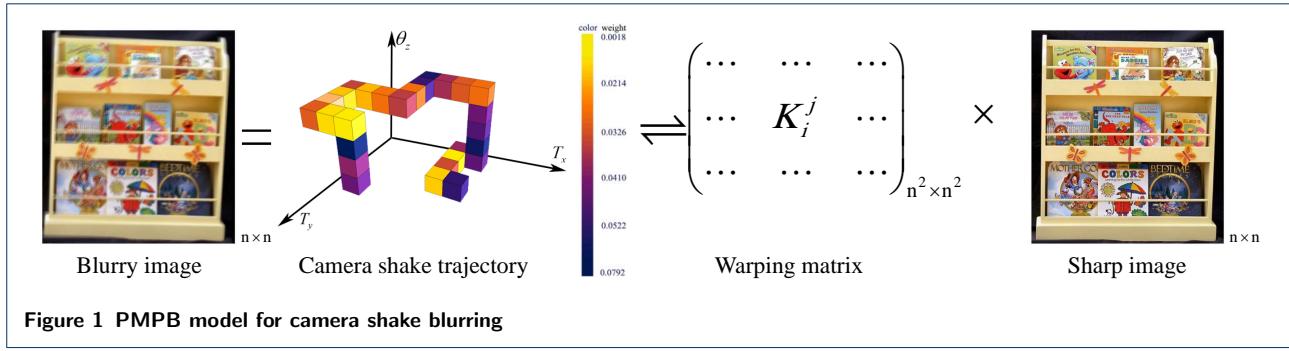
In this paper, we propose generalized additive convolution (GAC) model, by which a camera motion trajectory can be decomposed into in-plane translations (slice) or roll rotations (fiber), resulting in slice GAC model or fiber GAC model. In slice GAC model, a homography is formulated as the rotation of the convolution image of slice and sharp image, and in fiber GAC model, a homography is the inverse polar transform of the convolution of a kernel with the transformed sharp image (Please refer to Section 3.2 for detailed proof), in which the convolution operation can be efficiently computed with the help of FFT. By this way, the GAC-based forward blur models only require several FFTs and a number of pixel-wise operations, significantly reducing computational complexity, and concurrently only the basis kernels and a number of rotation matrices are required to be stored so that the problem of memory burden is also relaxed. Furthermore, a greedy algorithm is proposed to generate a more compact set of slices and fibers from any camera motion trajectory, resulting in hybrid GAC model, a more promising way to concurrently solve the problems of computational inefficiency and memory burden. As for the optimization algorithm, we adopt the MAP framework to alternatively estimate non-uniform blur and latent image, in which generalized accelerated proximal gradient (GAPG) algorithm [33], a much efficient optimization algorithm for non-blind deconvolution, is employed for GAC-based non-uniform deblurring.

Moreover, in slice GAC, we introduce group sparsity, i.e.,  $l_{2,1}$ -norm, on the pose weight matrix along the rotation angle dimension, interestingly providing a way to make non-uniform deblurring approach toward uniform blind deconvolution. Under alternative minimization framework, we also propose an effective solution to solve this problem. With the group sparsity constraint, more slices with rotation angles around 0 can be activated, and extremely when all the slices except angle 0 are inactivated, the non-uniform deblurring method can degrade to uniform blind deconvolution.

Experimental results show that the proposed GAC method can obtain comparable or better deblurring results than the competing uniform and non-uniform deblurring methods. Compared to non-uniform deblurring method, GAC has a much lower peak memory usage than [16, 29, 32], and is much more efficient than the state-of-the-art camera shake removal methods [16, 28, 29, 31, 32].

We summarize our contributions as follows:

- (1) We develop generalized additive convolution (GAC) framework together with two GAC models, i.e., slice GAC and fiber GAC, for forward modeling



of camera shake blur, by which FFT can be employed for efficient computation.

- (2) To further reduce the computational complexity, a greedy algorithm is proposed to generate hybrid GAC from any camera motion trajectory, resulting in a more compact GAC model.
- (3) As for the optimization algorithm, we adopt a fast gradient method, i.e., GAPG [33], to estimate latent image, contributing to the faster convergence of the deblurring algorithm.
- (4) By imposing group sparsity of pose weight matrix in slice GAC, we interestingly provide a way to connect uniform and non-uniform deblurring.

The rest of the paper is organized as follows. Section 2 provides a brief review on camera shake models and optimization algorithms. Section 3 presents the proposed GAC models, which in Section 4 are embedded in non-uniform deblurring, and can be efficiently solved via GAPG. Section 5 presents experimental results, and finally Section 6 concludes this paper.

## 2 Prerequisites and Related Work

For efficient non-uniform deblurring, it is crucial to design forward blur models and efficient optimization algorithms. In this section, we briefly review the original projective motion path blur (PMPB) model and its simplified 3D approximation, and the optimization algorithms in MAP-based deblurring.

### 2.1 Homography based model

In the PMPB model, a camera shaken image is geometrically the integration of what the camera 'sees' along the motion trajectory that can be modeled in a 6D camera pose space including  $x$ -,  $y$ -,  $z$ -axes rotation and  $x$ -,  $y$ -,  $z$ -axes translation. With the discretion of the time  $t$  and camera pose space, the integration can be rewritten as the weighted summation of all homographies in the discrete camera pose space, where a homography is the transformed sharp image projected by corresponding camera pose.

For each pose  $j$  along the camera motion trajectory, the corresponding homography can be represented as:

$$\mathbf{H}_j = \mathbf{C} \left( \mathbf{R}_j + \frac{\mathbf{t}_j}{d} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \right) \mathbf{C}^{-1} \quad (1)$$

where  $\mathbf{C}$  is the matrix of camera intrinsic parameters,  $d$  is the depth of the scene,  $\mathbf{R}_j$  and  $\mathbf{t}_j$  are the rotation matrix and translation vector, respectively.

Given  $\mathbf{H}_j$ , we can construct the corresponding warping matrix  $\mathbf{K}_j$ . Fig. 1 shows a flowchart of PMPB model, the PMPB model treats the process of non-uniform blur as a summation over the images transformed by the warping matrix

$$\mathbf{b} = \sum_j w_j \mathbf{K}_j \mathbf{x} + \mathbf{v} = \mathbf{K} \mathbf{x} + \mathbf{v} \quad (2)$$

where  $w_j$  is the fraction of time the camera spent at pose  $j$ ,  $\mathbf{b} \in \mathbb{R}^{n \times n}$  is the blurry image,  $\mathbf{x} \in \mathbb{R}^{n \times n}$  is the latent sharp image,  $\mathbf{K}$  is the sparse warping matrix with size  $n^2 \times n^2$ , and  $\mathbf{v}$  denotes the additive Gaussian white noise.

PMPB is faithful to real world camera motion, but has too many unknowns to be estimated. Fortunately, recent studies have shown that the full 6D camera pose space can be well approximated by the discrete 3D subspace of pitch, yaw, and roll rotations [28], or the 3D subspace of in-plane translations and roll rotation in wide focal lengths range [29], which are both effective [34]. In [32], constrained camera pose subspace was further introduced to refine the set of camera poses. Interestingly, the global camera motion constraint can be adopted to guide the construction of local blur kernels [31]. However, the method in [28] suffers from the problem of heavy computational load to compute huge warping matrices, while the methods in [29, 32] suffer from the extensive memory burden problem to store huge warping matrices.

### 2.2 Additive convolution model

Deng et al. [35] suggested an additive convolution (AC) model for non-blind deblurring, where non-uniform

blur is modeled as the weighted summation of the convolution of the sharp image with a set of basis convolution kernels

$$\mathbf{b} = \sum_{i=1}^C \boldsymbol{\alpha}_i \circ (\mathbf{x} \otimes \mathbf{k}_i) \quad (3)$$

where  $\mathbf{k}_i$  is the  $i$ -th basis convolution kernel,  $\boldsymbol{\alpha}_i$  denotes the  $i$ -th weight matrix,  $\otimes$  denotes the convolution operation,  $\circ$  denotes the pixel-wise multiplication operator, and  $C$  is the number of basis convolution kernels. With FFT, the computational complexity of AC model is only  $O(Cn^2 \log n)$ . Furthermore, principal component analysis (PCA) is adopted to learn the basis convolution kernels in advance. However, when applied to blind non-uniform deblurring, both the basis kernels and the weighted maps should be updated in each iteration, making it unpractical in blind deblurring of camera shaken images.

In this paper, we propose a fast forward blur model, i.e., GAC, to represent non-uniform blur caused by camera shake, which is distinctly different with the existing models. Instead of utilizing sparse warping matrix [28, 29, 32], GAC models reformulate homography as convolution related to slice and fiber, and can achieve better trade-off between computational cost and memory complexity. Different from [31], GAC is an efficient implementation of the global geometric model rather than the locally uniform approximation. Compared with [35], the basis convolution kernels and the weighted maps can be efficiently constructed based on the camera motion trajectory, and thus is easy to update.

### 2.3 MAP-based Blind Deblurring

Given the blurry image  $\mathbf{b}$ , MAP strategy is usually adopted to estimate the blur  $\mathbf{K}$  and the latent sharp image  $\mathbf{x}$  by minimizing the negative logarithm of posterior probability with respect to  $\mathbf{K}$  and  $\mathbf{x}$ ,

$$\begin{aligned} -\log p(\mathbf{K}, \mathbf{x} | \mathbf{b}) &\propto -\log p(\mathbf{y} | \mathbf{K}, \mathbf{x}) - \log p(\mathbf{x}) \\ &\quad - \log p(\mathbf{K}) \\ &\propto \frac{1}{2} \|\mathbf{Kx} - \mathbf{b}\|^2 + \Phi_1(\mathbf{x}) + \Phi_2(\mathbf{K}) \end{aligned} \quad (4)$$

where  $\Phi_1(\mathbf{x})$  denotes the regularizer on the latent sharp image, and  $\Phi_2(\mathbf{K})$  denotes the regularizer on the blur operator. For non-uniform blur,  $\mathbf{Kx} = \sum_j w_j \mathbf{K}_j \mathbf{x}$ , where the blur operator is parameterized by the weights  $w_j$ . Like the alternative updating strategy commonly adopted in uniform blind deconvolution [16, 18], the estimation of weight matrix  $\mathbf{W}$  and latent image  $\mathbf{x}$  should be alternatively performed.

To constrain the ill-posed problem for better deblurring quality, proper regularizers should be imposed. In

3D camera pose subspace, only a few poses are active given a camera motion trajectory so that it is reasonable to introduce sparsity on  $\mathbf{W}$ . In [28, 32],  $l_2$ -norm of weight matrix or its gradient is imposed, where the estimation of  $\mathbf{W}$  is a linear least square problem, which can be solved by gradient based optimization method, while in [29], non-convex regularizer is introduced, which is optimized by an iterative re-weighted least squares (IRLS).

As for updating the latent clear image, the edge prediction step is usually necessary to guarantee the algorithm converge to the desired solution, so that the sparsity, e.g., total variation [28, 32], is also imposed. In earlier researches, simple but efficient optimization method, e.g., Richardson-Lucy (RL) algorithm, was adopted to solve the problem [27], however the deblurring results often suffer from ringing artifacts. As a non-blind deconvolution problem, fast gradient-based optimization method has been intensively studied [36, 37]. The iterative shrinkage thresholding algorithm (IST) [36] was first proposed, and due to its simplicity and efficiency, two accelerated IST-based algorithms had been developed, i.e., FISTA [38] and TwIST [37], which both possess higher convergence rate. Furthermore, Zuo and Lin proposed GAPG algorithm [33], which further accelerated the convergence rate of APG algorithm.

In this paper, the GAC-based deblurring problem is solved in the alternative minimization framework, where the updating of pose weight matrix with both sparsity and group sparsity regularizers can be solved, and the updating of latent clear image can be efficiently solved by GAPG algorithm.

## 3 Generalized Additive Convolution Model for Camera Shake

In this section, we first propose the general form of the GAC model, and then decompose camera motion trajectory into slices and fibers, which provides a solution to specify GAC model for efficient modeling of camera shake blur. Finally, we propose a greedy algorithm to generate hybrid GAC model, and discuss the memory and computational complexity of the GAC model.

### 3.1 Generalized Additive Convolution Model

The form of the generalized additive convolution (GAC) model is defined as

$$\mathbf{b} = \sum_{i=1}^C \mathbf{f}_i(\mathbf{g}_i(\mathbf{x}) \otimes \mathbf{k}_i) \quad (5)$$

where  $\mathbf{f}_i$  and  $\mathbf{g}_i$  are two pixel-wise operators. GAC is the generalization of the AC model [35], by defining  $\mathbf{f}_i(\mathbf{x}) = \boldsymbol{\alpha}_i \circ \mathbf{x}$  and  $\mathbf{g}_i(\mathbf{x}) = \mathbf{x}$ . Moreover, since both

$f_i$  and  $g_i$  are pixel-wise operators, the computational complexity of GAC is the same as that of the AC model, i.e.,  $O(Cn^2\log_2 n)$ , where  $n^2$  is the number of pixels of the image  $\mathbf{x}$ . The smaller the  $C$  value indicates that GAC model is more efficient. Thus, the key issue of GAC model is to better specify  $k_i$ ,  $f_i$  and  $g_i$  to reduce the  $C$  value.

### 3.2 Decomposition of slice and fiber

In this subsection, we show that camera poses in the 3D camera pose subspace [29] can be decomposed into slices and fibers, which then provides proper choices to design  $k_i$ ,  $f_i$  and  $g_i$  in GAC model to reduce  $C$  value.

In the 3D camera pose subspace [29], pose in a camera motion trajectory is parameterized as,

$$\boldsymbol{\theta} = (\theta_z, t_x, t_y) \quad (6)$$

where  $\theta_z$  is the roll angle,  $t_x$  and  $t_y$  are the translations along  $x$ - and  $y$ -axes, respectively. For each pose in the 3D camera subspace  $\boldsymbol{\theta}_j = (\theta_{z,j}, t_{x,j}, t_{y,j})$ , by defining,

$$\mathbf{M}_{\boldsymbol{\theta}_{j,r}} = \begin{bmatrix} \cos \theta_{z,j} & -\sin \theta_{z,j} & t_{x,j} \\ \sin \theta_{z,j} & \cos \theta_{z,j} & t_{y,j} \\ 0 & 0 & 1 \end{bmatrix} \quad (7)$$

the homography [39] can be defined as:

$$\begin{aligned} \mathbf{H}_{\boldsymbol{\theta}_j} &= \mathbf{C} \mathbf{M}_{\boldsymbol{\theta}_j} \mathbf{C}^{-1} \\ &= \mathbf{C} \left( \mathbf{R}_{\theta_{z,j}} + \frac{1}{d} * \mathbf{t}_j \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \right) \mathbf{C}^{-1} \\ &= \mathbf{C} \begin{bmatrix} \cos(\theta_{z,j}) & -\sin(\theta_{z,j}) & t_{x,j} \\ \sin(\theta_{z,j}) & \cos(\theta_{z,j}) & t_{y,j} \\ 0 & 0 & 1 \end{bmatrix} \mathbf{C}^{-1} \end{aligned} \quad (8)$$

where  $\mathbf{t}_j = [t_{x,j}, t_{y,j}, 1]^T$  is the translation vector,  $d$  is the depth of the scene.

As in [29, 32], we also assume that the camera intrinsic parameters are known in advance, and the calibration matrix has the standard form:

$$\mathbf{C} = \begin{vmatrix} \alpha_x & 0 & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{vmatrix} \quad (9)$$

where  $\alpha_x = f * \frac{m_{x\_ccd}}{m_{x\_im}}$  is scale factors relating pixels to distance,  $m_{x\_ccd}$  is the maximum width of the CCD,  $m_{x\_im}$  is the maximum width of the related image,  $\alpha_y = f * \frac{m_{y\_ccd}}{m_{y\_im}}$  is quantified in the same manner,  $f$  is the focal length, and  $(x_0, y_0)$  is the centre coordinate.

Given  $\mathbf{H}_{\boldsymbol{\theta}_j}$ , we can construct the corresponding warping matrix  $\mathbf{K}_{\boldsymbol{\theta}_j}$  (More detailed explanations can be found in [29, 32, 35]). We then have that the homography transform can be decomposed into in-plane translation followed by roll rotation, yielding the following proposition.

### Proposition 1

$$\mathbf{K}_{\boldsymbol{\theta}_j} = \mathbf{K}_{\boldsymbol{\theta}_{j,r}} \mathbf{K}_{\boldsymbol{\theta}_{j,t}}$$

Proof By defining

$$\mathbf{R}_{\boldsymbol{\theta}_{j,r}} = \begin{bmatrix} \cos \theta_{z,j} & -\sin \theta_{z,j} & 0 \\ \sin \theta_{z,j} & \cos \theta_{z,j} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}_{\boldsymbol{\theta}_{j,t}} = \begin{bmatrix} 1 & 0 & t_{x,j} \\ 0 & 1 & t_{y,j} \\ 0 & 0 & 1 \end{bmatrix}$$

and according to the definition of  $\mathbf{M}_{\boldsymbol{\theta}_j}$  in Eq. (7) it is obvious to see that  $\mathbf{M}_{\boldsymbol{\theta}_j} = \mathbf{T}_{\boldsymbol{\theta}_{j,t}} \mathbf{R}_{\boldsymbol{\theta}_{j,r}}$ . Based on the definition of  $\mathbf{H}_{\boldsymbol{\theta}_j}$  in Eq. (8), we can define  $\mathbf{H}_{\boldsymbol{\theta}_{j,r}}$  and  $\mathbf{H}_{\boldsymbol{\theta}_{j,t}}$  as  $\mathbf{H}_{\boldsymbol{\theta}_{j,r}} = \mathbf{C} \mathbf{R}_{\boldsymbol{\theta}_{j,r}} \mathbf{C}^{-1}$  and  $\mathbf{H}_{\boldsymbol{\theta}_{j,t}} = \mathbf{C} \mathbf{T}_{\boldsymbol{\theta}_{j,t}} \mathbf{C}^{-1}$ . One can easily see that,

$$\begin{aligned} \mathbf{H}_{\boldsymbol{\theta}_{j,t}} \mathbf{H}_{\boldsymbol{\theta}_{j,r}} &= \mathbf{C} \mathbf{T}_{\boldsymbol{\theta}_{j,t}} \mathbf{C}^{-1} \mathbf{C} \mathbf{R}_{\boldsymbol{\theta}_{j,r}} \mathbf{C}^{-1} \\ &= \mathbf{C} \mathbf{T}_{\boldsymbol{\theta}_{j,t}} \mathbf{R}_{\boldsymbol{\theta}_{j,r}} \mathbf{C}^{-1} = \mathbf{H}_{\boldsymbol{\theta}_j} \end{aligned}$$

and we then have

$$\mathbf{H}_{\boldsymbol{\theta}_j}^{-1} = \mathbf{H}_{\boldsymbol{\theta}_{j,r}}^{-1} \mathbf{H}_{\boldsymbol{\theta}_{j,t}}^{-1} \quad (10)$$

Based on the definition of  $\mathbf{K}_{\boldsymbol{\theta}_j} \mathbf{x}$ , for the pixel at the location  $[l_{x1}, l_{x2}, 1]^T$ ,  $\mathbf{K}_{\boldsymbol{\theta}_j} \mathbf{x}$  assigns the value of the pixel located at  $\mathbf{H}_{\boldsymbol{\theta}_j}^{-1}([l_{x1}, l_{x2}, 1]^T)$  of  $\mathbf{x}$  to the pixel located at  $[l_{x1}, l_{x2}, 1]$  of  $\mathbf{K}_{\boldsymbol{\theta}_j} \mathbf{x}$ . Based on Eq. (10),  $\mathbf{H}_{\boldsymbol{\theta}_j}^{-1}([l_{x1}, l_{x2}, 1]^T)$  can also be explained as  $\mathbf{H}_{\boldsymbol{\theta}_{j,r}}^{-1} \mathbf{H}_{\boldsymbol{\theta}_{j,t}}^{-1}([l_{x1}, l_{x2}, 1]^T)$ , which means that  $\mathbf{K}_{\boldsymbol{\theta}_j} \mathbf{x} = \mathbf{K}_{\boldsymbol{\theta}_{j,r}} \mathbf{K}_{\boldsymbol{\theta}_{j,t}} \mathbf{x}$ .  $\square$

Proposition 1 indicates that,  $\mathbf{K}_{\boldsymbol{\theta}_j} \mathbf{x}$  is the combination of two atom operations, i.e., first translating the image  $\mathbf{x}$  by  $t_{x,j}$  and  $t_{y,j}$  along  $x$ -axis and  $y$ -axis respectively, and then rotating the translated image by roll angle  $\theta_{z,j}$ . So we can rewrite  $\mathbf{K}_{\boldsymbol{\theta}_{j,r}} \mathbf{x}$  and  $\mathbf{K}_{\boldsymbol{\theta}_{j,t}} \mathbf{x}$  as:

$$\mathbf{K}_{\boldsymbol{\theta}_{j,r}} \mathbf{x} = \mathbf{R}_{\theta_{z,j}}(\mathbf{x}) \quad (11)$$

$$\mathbf{K}_{\boldsymbol{\theta}_{j,t}} \mathbf{x} = \mathbf{k}_{\mathbf{t}_j} \otimes \mathbf{x} \quad (12)$$

where  $\mathbf{R}_{\theta_{z,j}}$  denotes the pixel-wise image rotation operation, and the translation convolution kernel is defined as:

$$\mathbf{k}_{\mathbf{t}_j}(x, y) \begin{cases} 1, & \text{if } x = t_{x,j} \text{ and } y = t_{y,j} \\ 0, & \text{else} \end{cases} \quad (13)$$

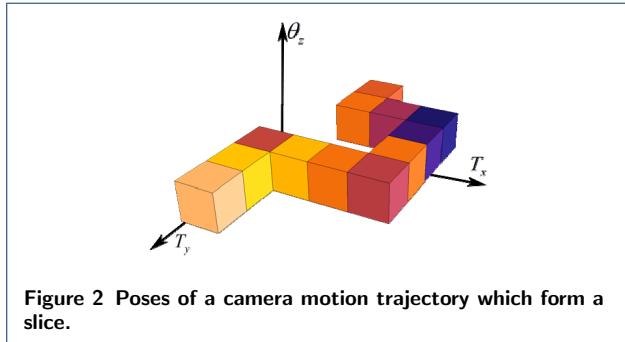
The non-uniform forward blur model Eq. (2) can be further formulated as:

$$\mathbf{K}\mathbf{x} = \sum_{\theta_j} w_{\theta_j} \mathbf{K}_{\theta_j} \mathbf{x} = \sum_{\theta_j} w_{\theta_j} \mathbf{K}_{\theta_j, r} \mathbf{K}_{\theta_j, t} \mathbf{x} \quad (14)$$

where  $w_{\theta_j}$  denotes the contribution (weight) of pose  $\theta_j$ . It should be noted that, the poses along a real camera motion trajectory form a connected 1D path, and thus the weights of most poses are zeros. So in Eq. (14) we can only consider the subset of poses with positive weights,  $\mathcal{P} = \{\theta_j : w_{\theta_j} > 0\}$ .

Finally, we define two special classes of subsets of  $\mathcal{P}$ : slice and fiber. A slice  $\mathcal{S}_\theta$  is defined as  $\mathcal{S}_\theta = \{\theta_j \in \mathcal{P} \text{ and } \theta_{z,j} = \theta\}$ , while a fiber  $\mathcal{F}_t$  is defined as  $\mathcal{F}_t = \{\theta_j \in \mathcal{P} \text{ and } (t_{x,j}, t_{y,j}) = t\}$ . Actually,  $\mathcal{P}$  can be decomposed into a number non-intersected slices and fibers. In the following, we will introduce how to construct slices and fibers from  $\mathcal{P}$  and how to reformulate camera shake as the GAC model.

### 3.3 Slice based GAC



**Figure 2** Poses of a camera motion trajectory which form a slice.

Figure 2 shows an example of a slice of camera poses which have the same roll angle. Given a slice  $\mathcal{S}_\theta$ , the blur caused by camera motion within  $\mathcal{S}_\theta$  can be formulated as

$$\begin{aligned} \mathbf{K}_\theta \mathbf{x} &= \sum_{\theta_j \in \mathcal{S}_\theta} w_{\theta_j} \mathbf{K}_{\theta_j, r} \mathbf{K}_{\theta_j, t} \mathbf{x} \\ &= \sum_{\theta_j \in \mathcal{S}_\theta} w_{\theta_j} \mathbf{R}_\theta (\mathbf{k}_{\theta_j} \otimes \mathbf{x}) \\ &= \mathbf{R}_\theta \left( \left( \sum_{\theta_j \in \mathcal{S}_\theta} w_{\theta_j} \mathbf{k}_{\theta_j} \right) \otimes \mathbf{x} \right) \\ &= \mathbf{R}_\theta (\mathbf{k}_{\mathcal{S}_\theta} \otimes \mathbf{x}) \end{aligned} \quad (15)$$

where  $\mathbf{k}_{\mathcal{S}_\theta}$  denotes the slice kernel with respect to the roll angle  $\theta$ .

For a general camera motion trajectory  $\mathcal{P}$ , we first classify the poses in  $\mathcal{P}$  into a number of non-intersected

slices with  $\mathcal{P} = \cup_\theta \{\mathcal{S}_\theta\}$ , and then the non-uniform blur in Eq. (14) can be equivalently reformulated as

$$\mathbf{K}\mathbf{x} = \sum_{\theta} \mathbf{R}_\theta (\mathbf{k}_{\mathcal{S}_\theta} \otimes \mathbf{x}) \quad (16)$$

It is obvious that this is a GAC model with  $\mathbf{f}_\theta(\mathbf{x}) = \mathbf{R}_\theta(\mathbf{x})$  and  $\mathbf{g}_\theta(\mathbf{x}) = \mathbf{x}$ . If the range of the roll angles is discretized into  $n_z$  intervals, we can see that the number  $C$  in slice GAC should be not higher than  $n_z$ .

Similarly, we can define the adjoint operator  $\mathbf{K}^T \mathbf{y}$  as

$$\mathbf{K}^T \mathbf{y} = \sum_{\theta} \tilde{\mathbf{k}}_{\mathcal{S}_\theta} \otimes \mathbf{R}_\theta^T(\mathbf{y}) \quad (17)$$

where  $\tilde{\mathbf{k}}_{\mathcal{S}_\theta}$  is the adjoint operator of  $\mathbf{k}_{\mathcal{S}_\theta}$  constructed by flipping the  $\mathbf{k}_{\mathcal{S}_\theta}$  upside-down and left-to-right, and  $\mathbf{R}_\theta^T$  is the strict adjoint operator of the discrete version of  $\mathbf{R}_\theta$ . It is obvious that the adjoint operator in Eq. (17) is also a GAC model and can be efficiently computed.

Finally, we discuss some implementation issues of slice GAC. To implement  $\mathbf{R}_\theta$ , one can simply adopt the Matlab command *imrotate*. To enhance the efficiency, we maintain a lookup table (LUT) for each discrete roll angle to record the correspondence of the coordinates before and after rotation. By discretizing the range of roll angles into  $n_z$  intervals, we pre-compute and store  $n_z$  LUTs. In the continuous case,  $\mathbf{R}_{-\theta}$  is the adjoint operator of  $\mathbf{R}_\theta$ , but in the discrete case, the error caused by discretization and interpolation cannot be overlooked. Thus, instead of using  $\mathbf{R}_{-\theta}$ , we adopt the strictly adjoint operator  $\mathbf{R}_\theta^T$ .

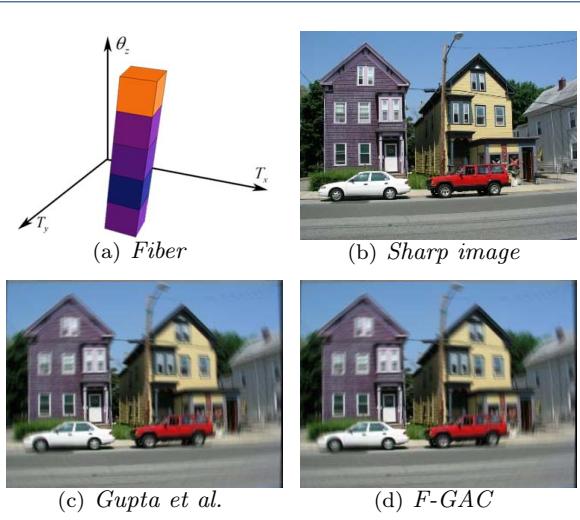
### 3.4 Fiber based GAC

Figure 3 shows an example of a fiber of camera poses which have the same translation  $\mathbf{t} = [t_x, t_y]^T$ , which applies a visual validation of our fiber based GAC model. Given a fiber  $\mathcal{F}_t$ , the non-uniform blur caused by the camera motion along  $\mathcal{F}_t$  can be formulated as

$$\begin{aligned} \mathbf{K}_t \mathbf{x} &= \sum_{\theta_j \in \mathcal{F}_t} w_{\theta_j} \mathbf{K}_{\theta_j, r} \mathbf{K}_{\theta_j, t} \mathbf{x} \\ &= \sum_{\theta_j \in \mathcal{F}_t} w_{\theta_j} \mathbf{R}_{\theta_{z,j}} (\mathbf{K}_{\theta_j, t} \mathbf{x}) \\ &= IPT(\mathbf{w}_t \otimes PT(\mathbf{K}_{\theta_t} \mathbf{x})) \end{aligned} \quad (18)$$

where  $\theta_t = [0, t_x, t_y]^T$ ,  $\mathbf{K}_{\theta_t} \mathbf{x}$  denotes the in-plane translation operation,  $PT(\cdot)$  and  $IPT(\cdot)$  stand for the polar transform and inverse polar transform [40], respectively. In the polar transform, we use the same interval to discretize angular and roll angles, and thus the basis filter  $\mathbf{w}_t$  can be defined as,

$$\mathbf{w}_t = [w_{t,\theta_1}, w_{t,\theta_2}, \dots, w_{t,\theta_{n_z}}] \quad (19)$$



**Figure 3** Simulated blurry images obtained using fiber GAC model Eq. (20) and using the 3D camera pose subspace model in [29]. (a) fiber, (b) sharp image, (c) blurry image based on the model in [29], and (d) blurry image based on fiber GAC model.

where  $\theta_1$  is the minimal roll angle and  $\theta_{nz}$  is the maximal roll angle.

We compare the model in Eq. (18) with the 3D camera pose subspace model in [29] for the simulation of blur caused by camera shake along a fiber. Figure 3 shows the blurry images obtained by using these two methods. One can easily see that the difference between Figs. 3(c) and Figs. 3(d) is insignificant, which demonstrates that the model in Eq. (18) can be used for modeling camera shaken image with a series of roll rotations.

For a general camera motion trajectory  $\mathcal{P}$ , we first classify the poses in  $\mathcal{P}$  into a number of non-intersected fibers with  $\mathcal{P} = \cup_t \{\mathcal{F}_t\}$ , and then the non-uniform blur in Eq. (14) can be equivalently reformulated as

$$\mathbf{K}\mathbf{x} = \sum_t \mathbf{K}_t \mathbf{x} = \sum_t IPT(\mathbf{w}_t \otimes PT(\mathbf{K}_{\theta_t} \mathbf{x})) \quad (20)$$

It is obvious that this is a GAC model with  $\mathbf{f}_t(\mathbf{x}) = IPT(\mathbf{x})$  and  $\mathbf{g}_t(\mathbf{x}) = PT(\mathbf{K}_{\theta_t} \mathbf{x})$ . If the range of the in-plane translations is discretized into  $n_x \times n_y$  intervals, then the number  $C$  in fiber GAC should be not higher than  $n_x n_y$ .

We can then define the adjoint operator  $\mathbf{K}^T \mathbf{y}$  as

$$\mathbf{K}^T \mathbf{y} = \sum_t \mathbf{K}_{-\theta_t} (PT^T(\tilde{\mathbf{w}}_t \otimes IPT^T(\mathbf{y}))) \quad (21)$$

where  $\tilde{\mathbf{w}}_t$  is the adjoint operator of  $\mathbf{w}_t$ ,  $PT^T$  and  $IPT^T$  are the adjoint operators of  $PT$  and  $IPT$ , respectively. To enhance the computational efficiency, two extra LUTs are pre-computed to record the cor-

respondence of polar and inverse polar transform, respectively. Moreover, we use the strict adjoint operators of  $PT$  and  $IPT$ , i.e.,  $PT^T$  and  $IPT^T$ , to avoid the inconsistency caused by discretization and interpolation.

### 3.5 Hybrid GAC for Modeling

For GAC, the key to save computational cost is to reduce  $C$ , the number of basis filters. Given a general camera motion trajectory as shown in Fig. 4, neither pure slice-based nor pure fiber-based GAC can guarantee sufficiently small value of  $C$ . However, in a hybrid (slice and fiber mixed) decomposition, only 2 slices and 2 fibers are required to model the camera motion trajectory, so that the computational complexity is significantly reduced. Thus, we propose a greedy method to decompose a camera motion trajectory into a hybrid set of slices and fibers to reduce the  $C$  value.

Given the pose subset  $\mathcal{P}$  and the 3D weight matrix  $\mathbf{W}$  with  $\mathbf{W}(\theta_z, t_x, t_y)$  the weight of the pose  $\theta = (\theta_z, t_x, t_y)$ , in each iteration, the proposed method first finds a candidate slice  $\mathcal{S}_{\hat{\theta}_z}$  and a candidate fiber  $\mathcal{F}_{\hat{t}_x, \hat{t}_y}$ , compare their relative contributions, and then choose the slice or fiber with higher weights. By this way, we can obtain a slice set  $\{(\theta_j, \mathcal{S}_j, \mathbf{k}_j) : j = 1, \dots, n_s\}$  and a fiber set  $\{(\mathbf{t}_i, \mathcal{F}_i, \mathbf{w}_i) : i = 1, \dots, n_f\}$ . As shown in Fig. 4, the proposed greedy algorithm can successfully decompose the camera motion trajectory into 2 slices and 2 fibers. The detailed algorithm is summarized in Algorithm 1.

Based on the slice set and the fiber set, the non-uniform caused by camera shake can be reformulated as:

$$\begin{aligned} \mathbf{K}\mathbf{x} = & \sum_{i=1}^{n_f} IPT(\mathbf{w}_i \otimes PT(\mathbf{K}_{\theta_{t_i}} \mathbf{x})) \\ & + \sum_{j=1}^{n_s} \mathbf{R}_{\theta_j}(\mathbf{k}_j \otimes \mathbf{x}) \end{aligned} \quad (22)$$

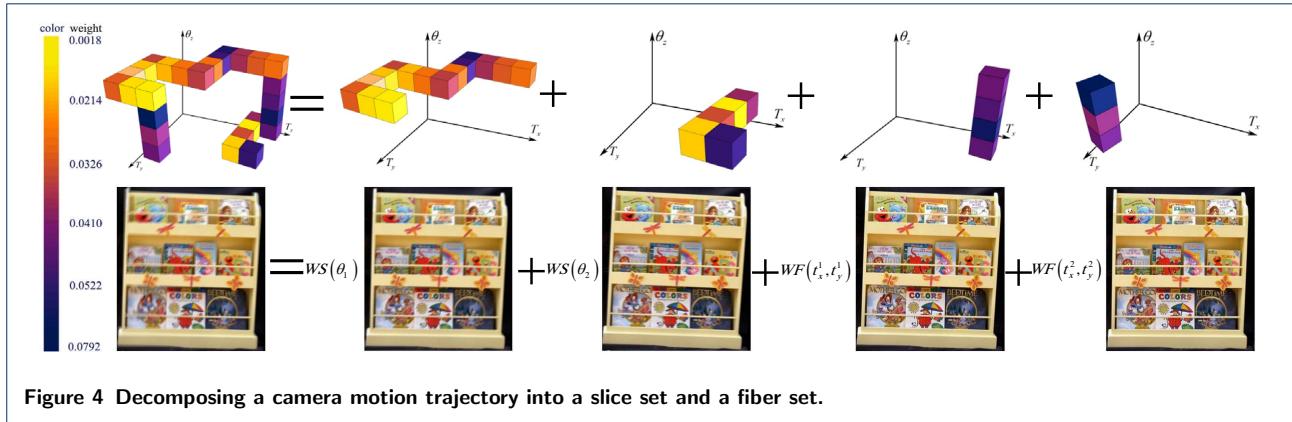
The adjoint operator  $\mathbf{K}^T \mathbf{y}$  is then defined as,

$$\begin{aligned} \mathbf{K}^T \mathbf{y} = & \sum_{i=1}^{n_f} \mathbf{K}_{-\theta_{t_i}} PT^T(\tilde{\mathbf{w}}_i \otimes IPT^T(\mathbf{y})) \\ & + \sum_{j=1}^{n_s} \tilde{\mathbf{k}}_j \otimes \mathbf{R}_{\theta_j}^T(\mathbf{y}) \end{aligned} \quad (23)$$

### 3.6 Discussions

With FFT, the computational complexity of the model in Eq. (22) is  $O((n_s + n_f)n^2 \log_2 n)$ . If  $n_s$  and  $n_f$  are small, GAC would be more efficient than the other methods. Let  $n_z$  be the number of intervals for the roll angle. It is reasonable to assume that  $(n_s + n_f) < n_z$ , otherwise, we can use the pure slice-based GAC model.

To further improve the computational efficiency, the LUT method can be adopted for fast image rotation,


**Algorithm 1** Greedy algorithm for hybrid GAC

**Input:**  $\mathcal{P}$  and  $\mathbf{W}$   
**Output:** The fiber set  $\{(t_i, \mathcal{F}_i, \mathbf{w}_j) : i = 1, \dots, n_f\}$  and the slice set  $\{(\theta_j, \mathcal{S}_j, \mathbf{k}_j) : j = 1, \dots, n_s\}$ .

- 1:  $\mathbf{WS}(\theta_z) = \sum_{t_x, t_y} \mathbf{W}(\theta_z, t_x, t_y)$ ,  $e = e_0 = \sum_{\theta_z} \mathbf{WS}(\theta_z)$ ,  $\mathbf{WF}(t_x, t_y) = \sum_{\theta_z} \mathbf{W}(\theta_z, t_x, t_y)$ ,  $n_s = 0$ ,  $n_f = 0$ ,  $\varepsilon, 0 < \alpha$ , null fiber set and null slice set.
- 2: **while**  $e/e_0 > \varepsilon$  **do**
- 3:    $(\hat{t}_x, \hat{t}_y) = \arg \max_{t_x, t_y} \mathbf{WF}(t_x, t_y)$  and  $w_f = \mathbf{WF}(\hat{t}_x, \hat{t}_y)$ .
- 4:    $\hat{\theta}_z = \arg \max_{\theta_z} \mathbf{WS}(\theta_z)$ ,  $w_s = \mathbf{WS}(\hat{\theta}_z)$ .
- 5:   Update the slice set: If  $w_s \geq \alpha w_f$ , add the  $\hat{\theta}_z^{th}$  slice to the slice set, and modify  $\mathbf{WS}(\hat{\theta}_z) = 0$ ,  $\mathbf{WF}(t_x, t_y) = \mathbf{WF}(t_x, t_y) - \mathbf{W}(\hat{\theta}_z, t_x, t_y)$ , and  $n_s = n_s + 1$ .
- 6:   Update the fiber set: If  $w_s < \alpha w_f$ , add the  $(\hat{t}_x, \hat{t}_y)^{th}$  fiber to the fiber set, modify  $\mathbf{WF}(\hat{t}_x, \hat{t}_y) = 0$ ,  $\mathbf{WS}(\hat{\theta}_z) = \mathbf{WS}(\hat{\theta}_z) - \mathbf{W}(\hat{\theta}_z, t_x, t_y)$ , and  $n_f = n_f + 1$ .
- 7:   Update  $e = \sum_{\theta_z} \mathbf{WS}(\theta_z)$ .
- 8:    $k \leftarrow k + 1$
- 9: **end while**

polar and inverse polar transform, and  $n_z + 2$  LUTs should be pre-computed and stored in memory. In [29, 32], a sparse  $n^2 \times n^2$  matrix was constructed for each pose in the 3D camera pose subspace. Compared with the model in [29, 32], GAC can achieve much better tradeoff between memory and computational complexity.

In [31], camera shake is approximated as the sum of  $R^2$  uniformly blurry patches, and the computational complexity of the model in [31] is  $O(R^2(n/R + w)^2 \log_2(n/R + w))$ , where  $w \times w$  is the size of the blur kernel. Compared with [31], when  $R$  and  $w$  are higher, GAC would be computationally more efficient, and our experimental results also validate the efficiency of GAC against the model by Hirsch et al. [31]. Moreover, the model in [31] is a locally uniform approximation of the camera shake model, while GAC can be strictly equivalent with the geometric model in [29].

## 4 GAC-based Non-uniform Deblurring

In this section, by incorporating the proposed GAC forward blur model into existing deblurring model, the GAC-based non-uniform deblurring can be efficiently solved. Then, by imposing  $l_{2,1}$ -norm regularizer on pose weight matrix  $\mathbf{W}$ , the slice GAC deblurring can approach toward uniform blind deconvolution.

### 4.1 GAC-based deblurring via GAPG

In typical non-uniform deblurring methods, TV regularizer on latent image and  $l_2$ -norm regularizer on pose weight matrix are imposed, yielding the deblurring formulation,

$$\min_{\mathbf{W}, \mathbf{x}} \left\| \sum_{\theta \in \mathcal{P}} w_\theta \mathbf{K}_\theta \mathbf{x} - \mathbf{b} \right\|^2 + \lambda \text{TV}(\mathbf{x}) + \tau \|\mathbf{W}\|^2 \quad (24)$$

where  $\lambda$  and  $\tau$  are trade-off parameters. By substituting the forward blur model with any GAC model, we come to the GAC-based non-uniform deblurring method, which can be solved by alternatively updating pose weight matrix  $\mathbf{W}$  and latent clear image  $\mathbf{x}$ . By fixing  $\mathbf{x}$ , we use the method in [32] to update  $\mathbf{W}$ , while by fixing  $\mathbf{W}$ , we develop an efficient solution to  $\mathbf{x}$  based on GAPG algorithm.

In [33], GAPG was developed to solve uniform non-blind deblurring, where the Lipschitz constant is generalized to diagonal matrix that can guarantee faster convergence rate. By introducing two auxiliary variables  $\mathbf{d}_h$  and  $\mathbf{d}_v$ , we reformulate the TV based model into the following equivalent problem,

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{d}_h, \mathbf{d}_v} \frac{1}{2} ( & \mu \|\mathbf{K}\mathbf{x} - \mathbf{b}\|^2 + \|\mathbf{d}_v - \mathbf{D}_v \mathbf{x}\|^2 \\ & + \|\mathbf{d}_h - \mathbf{D}_h \mathbf{x}\|^2 ) + \lambda \mu \|(\mathbf{d}_v, \mathbf{d}_h)\|_{TV} \end{aligned} \quad (25)$$

where  $\mu$  is the relaxation parameter,  $\mathbf{D}_h$  and  $\mathbf{D}_v$  are the horizontal and vertical gradient operators, respectively. According to [33], in each iteration, several subproblems should be solved for the updating of  $\mathbf{x}$ ,  $\mathbf{d}_h$

and  $\mathbf{d}_v$ , respectively. We use the same method as [33] for updating  $\mathbf{d}_h$  and  $\mathbf{d}_v$ , and choose the proper Lipschitz matrix for updating  $\mathbf{x}$ . For the updating of  $\mathbf{x}$ , the subproblem is formulated as

$$\begin{aligned} \min_{\mathbf{x}} & \frac{\lambda_{\max}}{2} \| \mathbf{x} - \{ \mathbf{y}_{\mathbf{x}} - \lambda_{\max}^{-1} [\mathbf{K}^T (\mathbf{K}\mathbf{y}_{\mathbf{x}} - \mathbf{b}) \\ & + \mathbf{D}_v^T (\mathbf{D}_v \mathbf{y}_{\mathbf{x}} - \mathbf{y}_{\mathbf{d}_v}) \\ & + \mathbf{D}_h^T (\mathbf{D}_h \mathbf{y}_{\mathbf{x}} - \mathbf{y}_{\mathbf{d}_h}) ] \} \| \end{aligned} \quad (26)$$

where  $\mathbf{y}_{\mathbf{x}}$ ,  $\mathbf{y}_{\mathbf{d}_v}$ ,  $\mathbf{y}_{\mathbf{d}_h}$  have the same definition as in [33]. Here  $\mathbf{K}^T (\mathbf{K}\mathbf{y}_{\mathbf{x}} - \mathbf{b})$  is computed based on Eq. 22 and Eq. 23. In [33],  $\lambda_{\max}$  is set based on the inequality  $\lambda_{\max} \leq (\sqrt{\mu} \|\mathbf{K}\|_2 + \sqrt{\eta} \|\mathbf{D}_h\|_2 + \sqrt{\eta} \|\mathbf{D}_v\|_2)^2$  with  $\eta = 1$ ,  $\|\mathbf{D}_h\|_2 \leq 2$ , and  $\|\mathbf{D}_v\|_2 \leq 2$  [33]. The diagonal Lipschitz matrix is

$$\mathbf{L}_f = \text{diag}(\lambda_{\max} \mathbf{I}, \eta \mathbf{I}, \eta \mathbf{I}) \quad (27)$$

By using smaller  $\lambda_{\max}$  and  $\eta$ , GAPG algorithm will converge faster. For non-uniform blur, we choose  $\|\mathbf{K}\|_2 = 1$  based on the following proposition.

### Proposition 2

$$\|\mathbf{K}\|_2 \leq 1$$

*Proof* In matrix analysis, we have

$$\|\mathbf{K}\|_2 \leq \sqrt{\|\mathbf{K}\|_1 \|\mathbf{K}\|_{\infty}}$$

where  $\|\cdot\|_1$  and  $\|\cdot\|_{\infty}$  denote the largest  $l_1$ -norm of the columns and rows, respectively. For the uniform blur procedure  $\mathbf{b} = \mathbf{K}\mathbf{x}$ , where each row of  $\mathbf{K}$  is a shifted version of blur kernel  $\mathbf{k}$ , Zuo and Lin had proved that the  $l_1$ -norm of the row of matrix  $\mathbf{K}$  corresponding to the  $(i, j)$ -th entry of  $\mathbf{b}$  is  $\|\mathbf{k}\|_1$  [33]. For the non-uniform blur matrix  $\mathbf{K}$ ,  $(i, j)$ -th entry of  $\mathbf{b}$  has its own blur kernel  $\mathbf{k}^{(i,j)}$ , and consequently  $\|\mathbf{K}\|_{\infty} = \max_{(i,j)} \|\mathbf{k}^{(i,j)}\|_1$ . By the non-negative constraint  $k_i \geq 0, \forall i$  and the normalization constraint  $\sum_i k_i = 1$ ,  $l_1$ -norm of any blur kernel is 1, i.e.,  $\|\mathbf{k}^{(i,j)}\|_1 = 1$ . Thus, we have  $\|\mathbf{K}\|_{\infty} = 1$ . Similarly,  $\|\mathbf{K}\|_1 = 1$  can be derived from [33]. Thus, the inequality is proved.  $\square$

The updating of subproblems should be performed several iterations. With the properly chosen Lipschitz matrix, each variable has its own Lipschitz constant, rather than the largest one adopted in APG algorithm, so that the GAPG-based deblurring method has a faster convergence rate.

### 4.2 Connection with uniform blind deconvolution

We impose group sparsity, i.e.,  $l_{2,1}$ -norm, on weight matrix  $\mathbf{W}$  along roll angle dimension to connect GAC-based non-uniform deblurring and uniform blind deconvolution,

$$\min_{\mathbf{W}, \mathbf{x}} \left\| \sum_{\theta \in \mathcal{P}} w_{\theta} \mathbf{K}_{\theta} \mathbf{x} - \mathbf{b} \right\|^2 + \lambda \text{TV}(\mathbf{x}) + \sum_{\theta_j} \tau_{\theta_j} \|\mathbf{W}_{\theta_j}\|^2 \quad (28)$$

where  $\tau_{\theta_j}$  is the trade-off parameter controlling the weight of all slices with angle  $\theta_j$ , and by setting  $\tau_{\theta_j=0} < \tau_{\theta_i}, \forall i > j$ , the slices with larger rotation angle will be gradually inactivated, and naturally the non-uniform deblurring will approach toward uniform blind deconvolution.

To solve this problem, we also adopt the alternative minimization strategy, where the updating of latent clear image shares the same solution via GAPG as Section 4.1. Due to the non-smoothness of  $l_{2,1}$ -norm regularizer, we propose an effective solution to solve it. By introducing auxiliary variable  $\mathbf{W}'$ , the problem can be reformulated as,

$$\min_{\mathbf{W}, \mathbf{W}'} \left\| \sum_{\theta \in \mathcal{P}} w_{\theta} \mathbf{K}_{\theta} \mathbf{x} - \mathbf{b} \right\|^2 + \sum_{\theta_j} \tau_{\theta_j} \|\mathbf{W}'_{\theta_j}\|^2 + \frac{\delta}{2} \|\mathbf{W} - \mathbf{W}'\|^2 \quad (29)$$

where  $\delta$  is positive penalty parameter. For the  $\mathbf{W}$ -subproblem, it is also a linear least square problem, which can be efficiently solved [32], and for the  $\mathbf{W}'$ -subproblem, it is a standard  $l_{2,1}$ -norm optimization problem, which can be solved by group shrinkage operator [41].

The slice GAC with group sparsity provides a natural connection of non-uniform deblurring and uniform deconvolution. When there are only camera in-plane translations, i.e., no roll rotation, the camera shaken blur will be uniform, and the slice GAC can play like the traditional uniform blind deconvolution method. Otherwise, roll rotations will cause the non-uniform blur, where slice GAC can play like non-uniform deblurring method.

Fig. 5 shows an deblurring example of the standard slice GAC (S-GAC) and slice GAC with group sparsity (GS-GAC) on a real camera shake blurred image. From the distribution of pose weights shown as Fig. 5(d), the group sparsity constrains the slices gathering at small rotation angles, where especially slices with angle 0 dominate, while in slice GAC the activated poses are more randomly distributed. Thus, for the blurry image with slight roll rotation shown as Fig. 5(a), GS-GAC performs much better than S-GAC, significantly suffering less artifacts.



**Figure 5 Comparison of standard S-GAC and GS-GAC.** (a) Original camera shake blurred image (b) Deblurring result by standard S-GAC method (c) Deblurring result by GS-GAC (d) Distribution of slice angles where the weights with the same rotation angle are summated.



**Figure 6 The results of different GAC models on three real camera shaken image.** For the close-ups, from top to bottom rows are the input real camera shake blurred images, deblurring results by F-GAC, deblurring results by S-GAC, deblurring results by H-GAC.

**Table 1** Running time (s) and Peak memory usage (GB) of different GAC models.

Image ID	Image Size	Time			Memory		
		F-GAC	S-GAC	H-GAC	F-GAC	S-GAC	H-GAC
Books	512 × 768	947	698	543	1.71	1.75	1.60
Butcher Shop	401 × 601	486	403	351	0.95	0.97	0.97
Statue	710 × 523	979	694	664	1.39	1.49	1.51

## 5 Experimental Results

In this section, we evaluate the performance of proposed GAC deblurring methods. First, three GAC

variants, i.e., slice based GAC (S-GAC), fiber based GAC (F-GAC) and hybrid GAC (H-GAC), are com-

pared in terms of computational time, memory usage and visual quality. Due to its superiority over S-GAC and F-GAC, H-GAC is then compared with the state-of-the-art geometric methods, i.e., Whyte et al. [28], Gupta et al. [29], and Hu et al. [32], and the state-of-the-art EFF-based methods, i.e., Hirsch et al. [31] and Xu et al. [16]. Finally, in the comparison with the state-of-the-art uniform deblurring methods [7, 9, 18, 21], slice GAC with group sparsity (GS-GAC) is adopted. All the experiments are conducted on a PC with Intel Core i5-2400 3.10 GHz CPU and 16G RAM memory, and the proposed method is implemented in Matlab.

### 5.1 Comparison of three GAC variants

On three real world camera shaken images shown as Fig. 8, three GAC variants are evaluated in terms of running time and memory usage. As shown in Table 1, the three GAC methods have comparable peak memory cost to each other, while in terms of running time, H-GAC is more efficient than both S-GAC and F-GAC, since the set of slices and fibers decomposed by H-GAC is usually more compact, thus having lower computational complexity.

From the restoration quality shown as Fig. 6, H-GAC and S-GAC perform better than F-GAC in recovering more details and achieving more visual plausible deblurring quality. Due to its superiority over both S-GAC and F-GAC, H-GAC is adopted in the following comparison with non-uniform deblurring methods.

### 5.2 Comparison with geometrical methods

We use five real world camera shaken images, shown in the top row of Fig. 7, to compare GAC with three geometrical methods, i.e., Whyte et al. [28], Gupta et al. [29], and Hu et al. [32], where their deblurring results are obtained by running source codes or executable programs provided by authors. Although GAC costs more memory than Whyte et al. [28] shown as Table 2, it is at least 100× faster, and in terms of deblurring quality shown as Fig. 7, GAC performs much better than Whyte et al. [28] in achieving more clear and plausible texture details. At the same time, Gupta et al. [29], Hu and Yang [32] and the proposed GAC model actually adopt the same 3D subspace to approximate the full 6D camera pose space, and consequently can obtain the similar deblurring results. Thus, it is more critical to evaluate their performance in terms of computational efficiency and memory usage. Tables 2 and 3 show that, GAC not only is at least 2.5x faster, but also significantly relax memory burden than Gupta et al. [29], Hu and Yang [32].

### 5.3 Comparison with non-geometrical methods

We further compare GAC with three non-geometrical methods proposed by Harmeling et al. [42], Hirsch et

al. [31] and Xu et al. [16]. Since neither source code or executable program of Harmeling et al. [42] and Hirsch et al. [31] is available, we collected the deblurring results from their papers or websites. The non-geometrical methods greatly rely on the reasonability of region division, and often sacrifice image details to smooth out possible artifacts at region boundaries. As shown in Fig. 9(a) and Fig. 9(b), GAC can achieve more visually plausible deblurring results, while the results by Harmeling et al. [42] and Hirsch et al. [31] are visually over-smoothed. Xu et al. [16] provides an executable program, so in Table 4 we report the CPU running time and memory usage comparison with Xu et al. [16] on several camera shaken blurry images shown in Fig. 8, from which one can see that GAC also performs better than Xu et al. As for deblurring quality shown as Fig. 9(c), GAC can obtain comparable if not superior deblurring result than Xu et al. [16]

### 5.4 Comparison with uniform deblurring methods

In the comparison experiments with the state-of-the-art uniform deblurring methods, i.e., Shan et al. [9], Fergus et al. [7], Cho et al. [18] and Xu et al. [21], we adopt GS-GAC, which performs better in handling spatially invariant blurry images. In Fig. 10, the first image is of non-uniform blur with slight roll rotations, from which one can see that GAC can achieve more visually plausible deblurring results, such as the statue's face is recovered more clearly, while the uniform deblurring methods usually suffer from ringing artifacts. Then, on two spatially invariant blurred images, GAC can also achieve satisfactory deblurring results, since the imposed group sparsity enforces the GAC method to play like uniform deblurring method.

## 6 Conclusions

In this paper, by designing generalized additive convolution (GAC) model to geometrically represent camera shaken blur, non-uniform deblurring is efficiently addressed. Since the slices and fibers decomposed from the camera motion trajectory can be formulated as convolution, all the GAC methods can exploit FFT for efficient optimization. Compared with the methods in [29, 32], the proposed method only needs several FFTs and to store several basis convolution kernels and look-up-tables in memory. By incorporating group sparsity into pose weight matrix, the GAC-based deblurring methods can also work like uniform blind deconvolution, better handling uniform blurry images. Compared with non-uniform deblurring methods, GAC method has a much lower peak memory usage, and is much more efficient. Compared with uniform deblurring methods, GAC method also can achieve satisfactory deblurring results.



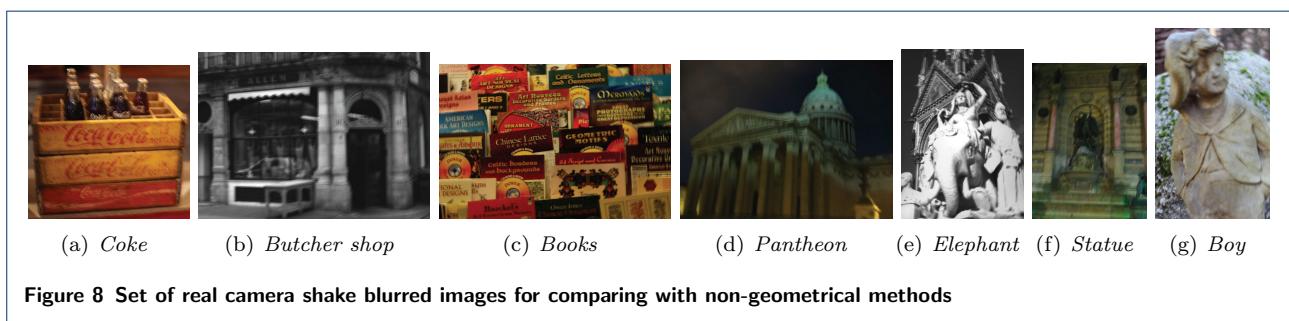
**Figure 7** Visual comparison of GAC model with three geometrical methods on five images. From top to bottom rows: blurry images, close-ups of blurry images, deblurring results of Whyte et al. [28], Gupta et al. [29], Hu and Yang [32], and GAC, respectively.

**Table 2** Running time (s) of the four geometrical methods.

Image ID	Image Size	Whyte et al.	Gupta et al.	Hu & Yang	GAC
Cabin	512 × 768	98872	7789	2113	514
Car-in-yard	512 × 768	87527	8829	7495	531
Dim-Petrol-Station	512 × 768	88767	8512	2512	596
Petrol-Station	406 × 679	89673	7982	1402	308
Six-Books	512 × 768	96732	7356	2601	572

**Table 3** Peak memory usage (GB) of the four geometrical methods.

Image ID	Image Size	Whyte et al.	Gupta et al.	Hu & Yang	GAC
Cabin	512 × 768	1.00	14.00	10.06	2.06
Car-in-yard	512 × 768	0.98	14.04	10.20	1.93
Dim-Petrol-Station	512 × 768	0.91	13.98	10.20	1.03
Petrol-Station	406 × 679	0.92	14.01	8.25	1.44
Six-Books	512 × 768	0.91	14.05	11.4	1.95



**Figure 8** Set of real camera shake blurred images for comparing with non-geometrical methods

## Acknowledgments

This work is supported in part by NSFC grants (61271093, 61471146, 61173086) and the program of ministry of education for new century excellent talents (NCET-12-0150).

## References

1. McCloskey, S., Ding, Y., Yu, J.: Design and estimation of coded exposure point spread functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34**(10), 2071–2077 (2012)
2. McCloskey, S.: Velocity-dependent shutter sequences for motion deblurring. In: *ECCV*, (2010)



**Figure 9** Deblurring effect comparison with non-geometrical methods. The top row from left to right are by Harmeling et al. [42], Hirsch et al. [31] and Xu et al. [16], respectively. The bottom row are deblurring results by GAC.

**Table 4** Running time (s) and peak memory usage (GB) comparison of Xu et al. [16] and hybrid GAC.

Image ID	Image Size	Time		Memory	
		Xu et al.	GAC	Xu et al.	GAC
Coke	680 × 731	1335	952	9.28	3.61
Butcher shop	401 × 601	922	351	2.24	1.50
Books	512 × 768	4698	596	2.94	2.03
Pantheon	273 × 366	260	69	0.73	0.45
Elephant	611 × 411	1099	418	1.87	1.03
Statue	710 × 523	1333	716	2.90	1.67
Boy	751 × 451	1017	708	1.80	1.35

3. Levin, A., Fergus, R., Durand, F., Freeman, W.T.: Image and depth from a conventional camera with a coded aperture. *ACM Trans. Graph.* **26**(3) (2007)
4. Joshi, N., Kang, S.B., Zitnick, C.L., Szeliski, R.: Image deblurring using inertial measurement sensors. *ACM Trans. Graph.* **29**(4), 30 (2010)
5. Tai, Y.-W., Du, H., Brown, M.S., Lin, S.: Correction of spatially varying image and video motion blur using a hybrid camera. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(6), 1012–1028 (2010)
6. Tai, Y.-W., Kong, N., Lin, S., Shin, S.Y.: Coded exposure imaging for projective motion deblurring. In: *CVPR*, pp. 2408–2415 (2010)
7. Fergus, R., Singh, B., Hertzmann, A., Roweis, S.T., Freeman, W.T.: Removing camera shake from a single photograph. *ACM Trans. Graph.* **25**(3), 787–794 (2006)
8. Levin, A., Weiss, Y., Durand, F., Freeman, W.T.: Understanding and evaluating blind deconvolution algorithms. In: *CVPR*, pp. 1964–1971 (2009)
9. Shan, Q., Jia, J., Agarwala, A.: High-quality motion deblurring from a single image. *ACM Trans. Graph.* **27**(3), 73 (2008)
10. Krishnan, D., Tay, T., Fergus, R.: Blind deconvolution using a normalized sparsity measure. In: *CVPR*, pp. 233–240 (2011)
11. Cho, T.S., Paris, S., Horn, B.K.P., Freeman, W.T.: Blur kernel estimation using the radon transform. In: *CVPR*, pp. 241–248 (2011)
12. Zhe, H., Ming-Hsuan, Y.: Good regions to deblur. In: *ECCV*, vol. 7576, pp. 59–72 (2012)
13. Miskin, J., MacKay, D.J.: Ensemble learning for blind image separation and deconvolution. In: *Advances in Independent Component Analysis*, pp. 123–141 (2000)
14. Levin, A., Weiss, Y., Durand, F., Freeman, W.T.: Efficient marginal likelihood optimization in blind deconvolution. In: *CVPR*, pp. 2657–2664 (2011)
15. Babacan, S.D., Molina, R., Do, M.N., Katsaggelos, A.K.: Bayesian blind deconvolution with general sparse image priors. In: *ECCV* vol. 7577, pp. 341–355 (2012)
16. Xu, L., Zheng, S., Jia, J.: Unnatural I/O sparse representation for natural image deblurring. In: *CVPR*, pp. 1107–1114 (2013)
17. Kotera, J., Šroubek, F., Milanfar, P.: Blind deconvolution using



**Figure 10 Deblurring effect comparison with uniform deblurring methods on uniformly blurred images.** First column is the real camera shake blurred image. From the second column, left to right are by Shan et al. [9], Fergus et al. [7], Cho et al. [18] and Xu et al. [21], respectively. The last column are deblurring results by GAC.

- alternating maximum a posteriori estimation with heavy-tailed priors. In: *Compt. Anal. Images Patterns*, vol. 8048, pp. 59–66 (2013)
18. Cho, S., Lee, S.: Fast motion deblurring. *ACM Trans. Graph.* **28**(5), 145 (2009)
  19. Joshi, N., Szeliski, R., Kriegman, D.: Psf estimation using sharp edge prediction. In: *CVPR*, pp. 1–8 (2008)
  20. Sun, L., Cho, S., Wang, J., Hays, J.: Edge-based blur kernel estimation using patch priors. In: *ICCP*, pp. 1–8 (2013)
  21. Xu, L., Jia, J.: Two-phase kernel estimation for robust motion deblurring. In: *ECCV*, pp. 157–170 (2010)
  22. Schelten, K., Nowozin, S., Jancsary, J., Rother, C., Roth, S.: Interleaved regression tree field cascades for blind image deconvolution. In: *IEEE Winter Conference on Applications of Computer Vision (WACV)* (2015)
  23. Schuler, C.J., Hirsch, M., Harmeling, S., Schölkopf, B.: Learning to deblur. *arXiv preprint arXiv:1406.7444* (2014)
  24. Zuo, W., Ren, D., Gu, S., Lin, L., Zhang, L.: Discriminative learning of iteration-wise priors for blind deconvolution. In: *CVPR* (2015)
  25. Hu, Z., Yang, M.-H.: Learning good regions to deblur images. *International Journal of Computer Vision*, 1–18 (2015)
  26. Ji, H., Wang, K.: A two-stage approach to blind spatially-varying motion deblurring. In: *CVPR*, pp. 73–80 (2012)
  27. Tai, Y.-W., Tan, P., Brown, M.S.: Richardson-lucy deblurring for scenes under a projective motion path. *IEEE Trans. Pattern Anal.* **Mach. Intell.** **33**(8), 1603–1618 (2011)
  28. Whyte, O., Sivic, J., Zisserman, A., Ponce, J.: Non-uniform deblurring for shaken images. *Int. J. Comp. Vis.* **98**(2), 168–186 (2012)
  29. Gupta, A., Joshi, N., Zitnick, C.L., Cohen, M., Curless, B.: Single image deblurring using motion density functions. In: *ECCV*, vol. 6311, pp. 171–184 (2010)
  30. Hirsch, M., Sra, S., Schölkopf, B., Harmeling, S.: Efficient filter flow for space-variant multiframe blind deconvolution. In: *CVPR*, pp. 607–614 (2010)
  31. Hirsch, M., Schuler, C.J., Harmeling, S., Schölkopf, B.: Fast removal of non-uniform camera shake. In: *ICCV*, pp. 463–470 (2011)
  32. Hu, Z., Yang, M.-H.: Fast non-uniform deblurring using constrained camera pose subspace. In: *BMVC* (2012)
  33. Zuo, W., Lin, Z.: A generalized accelerated proximal gradient approach for total-variation-based image restoration. *IEEE Trans. Image Process.* **20**(10), 2748–2759 (2011)
  34. Köhler, R., Hirsch, M., Mohler, B., Schölkopf, B., Harmeling, S.: Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database. In: *ECCV*, pp. 27–40 (2012)
  35. Deng, H., Zuo, W., Zhang, H., Zhang, D.: An additive convolution model for fast restoration of nonuniform blurred images. *Int. J. Comp. Math.* **91**(11), 2446–2466 (2014)
  36. Daubechies, I., Defrise, M., De Mol, C.: An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Comm.*

- Pure. Appl. Math. **57**(11), 1413–1457 (2004)
- 37. Bioucas-Dias, J.M., Figueiredo, M.A.: A new twist: two-step iterative shrinkage/thresholding algorithms for image restoration. IEEE Trans. Image Process. **16**(12), 2992–3004 (2007)
  - 38. Beck, A., Teboulle, M.: Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. IEEE Trans. Image Process. **18**(11), 2419–2434 (2009)
  - 39. Stančin, S., Tomažič, S.: Angle estimation of simultaneous orthogonal rotations from 3d gyroscope measurements. Sensors **11**(9), 8536–8549 (2011)
  - 40. Matungka, R., Zheng, Y.F., Ewing, R.L.: Image registration using adaptive polar transform. IEEE Trans. Image Process. **18**(10), 2340–2354 (2009)
  - 41. Yang, J., Yin, W., Zhang, Y., Wang, Y.: A fast algorithm for edge-preserving variational multichannel image restoration. SIAM Journal on Imaging Sciences **2**(2), 569–592 (2009)
  - 42. Harmeling, S., Hirsch, M., Schölkopf, B.: Space-variant single-image blind deconvolution for removing camera shake. In: NIPS, pp. 829–837 (2010)