

# Fast Non-uniform Deblurring using Constrained Camera Pose Subspace

Zhe Hu  
zhu@ucmerced.edu  
Ming-Hsuan Yang  
mhyang@ucmerced.edu

Electrical Engineering and Computer  
Science  
University of California at Merced

---

## Abstract

Camera shake during exposure time often results in non-uniform blur across the entire image. Recent algorithms model the non-uniform blurry image as a linear combination of images observed by the camera at discretized poses, and focus on estimating the time fraction positioned at each pose. While these algorithms show promising results, they nevertheless entail heavy computational loads. In this work, we propose a novel single image deblurring algorithm to remove non-uniform blur. We estimate the local blur kernels at different image regions and obtain an initial guess of possible camera poses using backprojection. By restraining the possible camera poses in a low-dimensional subspace, we iteratively estimate the weight for each pose in the camera pose space. Experimental validations with the state-of-the-art methods demonstrate the efficiency and effectiveness of our algorithm for non-uniform deblurring.

## 1 Introduction

Image deblurring is one of the most fundamental problems in computer vision that has been studied extensively. Image blur often results from relative motion between a camera and the scene (e.g., camera shake) during the exposure time. It causes significant image degradation, especially in the low light conditions where longer exposure time is required. Although the blur effect may be reduced by setting a faster shutter speed, it inevitably generates significant amount of noise.

Most existing algorithms formulate the image blur as a convolution process with a uniform blur kernel or point spread function (PSF) on a latent input image. The uniform blur model is sufficient if the camera shake is caused only by in-plane translation (i.e., within the sensor plane) without any out-of-plane rotations. However, this assumption does not hold in general. Camera tilts and rotations happen frequently and the image blur becomes non-uniform, i.e., blurs at different pixel locations are not the same. Moreover, object movements in the scene also lead to non-uniform blur effects. Thus, blur kernel estimation is especially difficult in the non-uniform case.

Early works address the non-uniform blur problem by restricting the blur process in simple cases [1, 2, 3]. Recently, a geometric model has been proposed to describe the camera motion using the weighted sum of homographically transformed copies of sharp image [4].

The transformed copies can be viewed as the intermediate images observed by the camera during the camera shake process. This model imposes a global constraint on local blur kernels, and the estimation step becomes computing the weights of different camera poses rather than directly recovering spatially varying blur kernels at different pixel locations. The main disadvantage of this model is the heavy computational load due to dense sampling of poses in the high dimensional camera motion space. To make this problem solvable in practice, recent methods approximate the six-dimensional motion space with a three-dimensional subspace (roll and in-plane translation) and coarse-grained sampling. Nevertheless, this simplification inevitably introduces reconstruction errors. Most recently, an approach that uses efficient filter flows to leverage the computational issues is proposed [10]. However, this algorithm aims to approximate local blur kernels without exploiting camera motion structure (e.g., camera motion trajectory is sparse in the camera motion space).

In this paper, we propose a single image deblurring algorithm to remove non-uniform blur. We first introduce an initialization method that facilitates convergence and avoid local minimums of the formulated optimization problem. We then propose a new camera motion estimation method which optimizes on a small set of pose weights of a constrained camera pose subspace at a time rather than using the entire space. We develop an iterative method to refine the camera motion estimation and introduce perturbation at each iteration to obtain robust solutions. The proposed method also enables us to use dense sampling and the six-dimensional motion space without heavy computational loads. We evaluate our method with the state-of-the-art algorithms and the experimental results show that the proposed approach is not only efficient but also effective for non-uniform image deblurring.

## 2 Related Work

The problem of removing spatially invariant blur has been studied extensively and numerous algorithms have been proposed [3, 5, 6, 7, 8, 14, 15, 16, 17, 20, 22]. Since blind deconvolution is an ill-posed problem, it is often necessary to exploit prior knowledge and additional information to obtain effective solutions. For single image deblurring, recent works focus on exploring image priors for deblurring. Image priors are usually formulated as constraints or regularization terms in the optimization function so that satisfactory results can be achieved without searching for the entire solution space. In [7], the heavy-tailed gradient distribution of natural images is exploited as prior information. The recovered latent image is constrained to fit the distribution of gradient magnitudes of natural images described by a Gaussian mixture. The underlying relation between motion blur and object boundary is analyzed and exploited for deblurring [12]. In [18], a deblurring algorithm is proposed in which prior knowledge regarding the gradient distribution of natural images is used along with consistency constraints of local smooth regions before and after blurring. The consistency constraints are shown to be effective in suppressing ringing effects. A method that uses sparsity constraints of both blur kernel and latent image is presented in [2]. Most recently, a fast method with carefully designed regularizations and optimization procedures is proposed for image deblurring [9].

Removing spatially variant blur is a challenging task in which regions are degraded differently, thereby entailing non-uniform models over the entire image. This problem has recently attracted more attention due to its wide range of practical applications. One approach to handle spatially varying blur is to segment the blurry image into multiple regions where each one is modeled by a uniform blur kernel [1, 2]. This approach is sensible and

well-known uniform image deblurring algorithms can be used once the segments are determined. Such segmented regions, however, often need to be small for the assumption of using uniform blur kernels to hold. Another approach is to model or capture the camera motion during the blur process and derive spatially varying blur kernels at different sites. In [14], an algorithm to account for blurs caused mainly by the in-plane rotational motion from a single image is proposed. Specially designed imaging systems (i.e., hybrid cameras and sensors) have also been used to estimate a local blur kernel at each pixel location based on optical flows [13, 15]. The use of additional hardware to facilitate image deblurring is demonstrated to be effective. Nevertheless, it is not clear how such approaches can be applied to single frame image deblurring in the general settings.

Recently, a geometric model has been proposed to model the observed blurry image as the integration of all the intermediate images captured by the camera along the motion trajectory [14, 15]. These intermediate images are modeled as the transformed ones (i.e., homographies) of the sharp image. Based on this model, image blur caused by camera motion can be well formulated with one optimization problem. It is also possible to remove spatially varying blur by estimating a general camera motion function. A similar model has been used to model three degrees of camera motion (with in-plane translation and rotation) in [9]. In [16], a method that employs an efficient filter flow algorithm to speed up the optimization step is developed. However, these approaches entail high computational cost as the high-dimensional camera motion and latent image have to be computed during the iterative optimization process.

### 3 Non-uniform Blur Model for Camera Shake

In this section, we describe the non-uniform blur model caused by camera shake for the scene with constant depth that can be modeled in a way similar to [9]. The geometric model we use is based on [16] which formulates the blurry image  $B$  as the integration of the transformed copies of latent image  $L$ ,

$$B = \int f(H_\theta, L)w(\theta)d\theta + n, \quad (1)$$

where  $\theta$  denotes the camera pose characterized by 6 parameters,  $H_\theta$  is the homography induced by camera pose  $\theta$ , and  $f(H_\theta, L)$  represents the transformed image. The blurry image can be viewed as a combination of the images the camera captures with observation noise  $n$ . Similar to [9], we address this problem in a discrete space with

$$B = \sum_{\theta \in S} f(H_\theta, L)w_\theta + n = \sum_{\theta \in S} w_\theta(K_\theta L) + n, \quad (2)$$

where  $K_\theta$  is the matrix that warps the latent image  $L$  to the transformed one at a sampled pose  $\theta$  and  $S$  denotes the set of sampled camera poses. In this formulation,  $K_\theta$  can also be viewed as a blur kernel basis where row  $i$  is the weight vector representing how pixel  $i$  of the latent image  $L$  contributes in the blurry image. We note that  $K_\theta$  is a sparse square matrix which can be stored in memory efficiently. Thus, we model the blur process as

$$B = \mathcal{K}L + n, \quad (3)$$

where  $\mathcal{K} = \sum_{\theta \in S} w_\theta K_\theta$  represents a blur matrix with each row as a point spread function (PSF) at a pixel location. For spatially invariant blur, each row of  $\mathcal{K}$  shares the same PSF but with location shifts. We denote  $W$  as the set of weights  $w_\theta$ .

To reduce the dimensionality, one approach is to approximate the six-dimensional camera pose space by three-dimensional rotations without considering translations [24]. In [9], it is shown that the out-of-plane rotation can be well approximated by  $x, y$  translation when the focal length is sufficiently long, which is usually the case for camera settings. In this work, we approximate the camera motion with a three-dimensional pose space as [9] ( $x, y$  translation and  $z$  rotation) and focus on estimation of pose weights.

## 4 Deblurring with Constrained Pose Subspace

The purpose of single image deblurring is to recover the sharp image  $L$  given the blurry observation  $B$ . To solve  $L$  in Eqn. 3, we seek to minimize the energy function over the variables  $L$  and  $W$  with kernel basis  $K_\theta$ ,

$$\min_{(L,W)} \left\| \left( \sum_{\theta \in \mathcal{S}} w_\theta K_\theta \right) L - B \right\|^2 + \Phi_1(L) + \Phi_2(W), \quad (4)$$

where  $\Phi_1$  and  $\Phi_2$  represent the regularization terms on the latent image  $L$  and weights  $W$  respectively. The regularization terms will be explained in the following sections.

In this work, we develop an iterative method to solve  $L$  and  $W$ . We first initialize  $W$  with patch-based uniform deblurred results, and then estimate  $L$  and  $W$  via iterative optimization. At each iteration, we restrict the solution space in a small set of camera pose space which the camera motion trajectory is most likely to belong to. Fig. 1 summarizes the main steps of our method and the technical details are presented in the following sections.

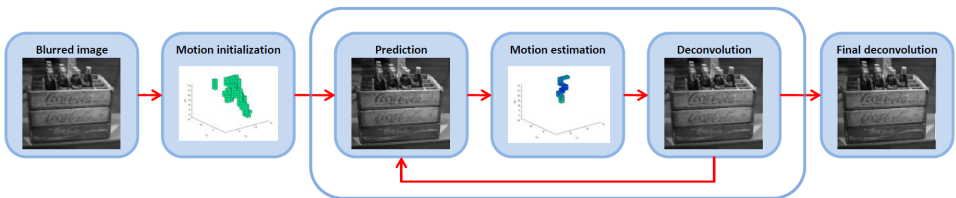


Figure 1: Main steps of the proposed algorithm.

### 4.1 Initialization using Backprojection

To solve the ill-posed problem, a good deconvolution method needs to deal with local minimums and multiple feasible solutions (pairs of sharp image and camera motion) for a given blurry image. Recent algorithms employ coarse-to-fine approaches to alleviate these problems, but require computing the kernel basis and estimating pose weights at each scale. In this work, we propose an initialization method of pose weights to address this problem. The proposed initialization method also facilitates the convergence of the optimization process.

The blur kernel at a specific pixel location can be viewed as the projection of the camera motion trajectory or a tomographic scan over the camera motion space. We denote  $p(\theta, l)$  as the mapping indicating how camera pose  $\theta$  affects pixel  $l$  in the image, which can also be viewed as the  $l$ -th row of kernel basis  $K_\theta$ . Thus, we express the blur kernel at site  $l$  as

$$k_l = \sum_{\theta} w_\theta p(\theta, l). \quad (5)$$

Motivated by the backprojection techniques in image processing [8] which are used to reconstruct the two-dimensional signal from its one-dimensional projections, we develop a method to reconstruct the camera motion from multiple blur kernels. We first collect the estimated blur kernels from several image patches (e.g., from 3 by 3 grid) using a fast uniform deblurring algorithm [9]. Although the blur kernels are not precisely estimated due to the camera rotational motion and the estimation errors, they capture the overall shape of the blur kernels (i.e., coarse camera motion) and thus can be used for initialization. From multiple estimated blur kernels of different regions, one approach to approximate the camera motion is to use the inverse transformation by duplicating the two-dimensional blur kernels across the camera motion space. That is, for each entry in  $k_l$ , we determine the possible camera poses whose projection  $p(\theta, l)$  at this site  $l$  is the interest entry and then duplicate the weight of this entry across all the possible camera poses. This procedure is called backprojection commonly used in image reconstruction [8] and we denote  $bp(k_l, l)$  as the backprojected value by

$$bp(k_l, l) = \sum_i \sum_{\{\theta | p(\theta, l) = i\}} k_l(i) \Gamma(\theta), \quad (6)$$

where  $i$  is an index for any location of blur kernel  $k_l$  and  $\Gamma(\theta)$  is an indicator function of camera poses with value 1 for pose  $\theta$  and 0 for others. If a pose of interest coincides with the camera motion trajectory (we refer this pose as inlier and outlier pose vice versa), its amplitude will be doubled ideally once two blur kernels at different locations are backprojected (which is known in image reconstruction from projections [8]). As the number of the blur kernels increases, the strength of outlier pose decreases with respect to the strength of inlier pose in which multiple backprojections intersect. In this way, we can approximate the weights  $W$  using backprojection  $\hat{W} = \frac{1}{n} \sum_l bp(k_l, l)$  with normalization term  $n$ .

The approach with direct backprojection may not render precise camera motion estimation for several reasons. First, a good approximation of camera motion requires a large number of blur kernels to rule out outlier poses. However, it is time consuming to compute a large number of blur kernels at different locations and thus we use only a few blur kernels for initialization. Second, the estimated blur kernel itself is not precise. Since we use a uniform deblurring method to estimate the blur kernels from image patches, the results are not accurate as in-plane camera rotation of camera shake is likely to occur. Within an image, the patches near the optical center are affected with less blur caused by in-plane camera rotation compared to those on the boundary.

To address these problems, we formulate the initialization estimation with an optimization problem and enforce sparse constraints of the weights to get better reconstruction results due to the fact that the camera motion trajectory is sparse in the motion space. Moreover, we assign each backprojection function  $bp(k_l, l)$  a confidence value  $a(l)$  based on the distance of the site  $l$  to the optical center which can usually be assumed to be the center of the image. Thus, the initial estimation of weights  $W$  is formulated as,

$$\hat{W} = \arg \min_W \|W - \sum_l a(l) bp(k_l, l)\|^2 + \|W\|_1, \quad (7)$$

where the confidence value needs to be normalized across all the sites, i.e.,  $a(l) \leftarrow \frac{a(l)}{\sum_m a(m)}$ . We show that the proposed initialization method provides a good approximation to the camera motion trajectory in Sec. 5.

## 4.2 Estimating Pose Weights

In the weight estimation step, we compute  $W$  given the current recovered latent image  $L$  based on Eqn. 4.

**Image Prediction:** To obtain more accurate estimates, we take a prediction step to approximate the latent image similar to [9]. In this step, a bilateral filter is first applied to suppress noise and small edges, and a shock filter is then used to restore strong edges (more details of the settings can be found in [9]). Each predicted image  $\tilde{L}$  has fewer artifacts and sharp edges. More importantly, the predicted image  $\tilde{L}$  resembles the latent image even when the estimated  $L$  from the previous iteration is blurry, and thus this step facilitates faster convergence.

**Weight Estimation on Constrained Pose Subspace:** Since image derivatives have been shown to be effective for reducing ringing effects in the optimization process for deblurring [9, 13], we formulate the weight estimation problem as

$$\min_W \sum_{\partial_*} \alpha_* \left\| \sum_{\theta \in S} w_\theta \partial_*(K_\theta \tilde{L}) - \partial_* B \right\|^2 + \Phi_2(W), \quad (8)$$

where  $\partial_* \in \{\partial_x, \partial_y, \partial_{xx}, \partial_{xy}, \partial_{yy}\}$  denotes the partial derivative in two directions of first and second order, and  $\alpha_* \in \{\alpha_1, \alpha_2\}$  is the weight for partial derivative of different order as [13].

In this formulation, a direct optimization on the whole pose set  $S$  is computationally expensive. We note that the discretized poses along the camera motion trajectory form a connected one-dimensional path in the camera motion space. Moreover, the one-dimensional path constitutes a sparse subset of the space. That is, the majority of poses are outliers (i.e., not on the trajectory) and their weights should be zero. In this work, we propose to optimize Eqn. 8 only on a sparse subset of poses referred as an active set  $A \subset S$ . The active set represents the set of poses which are likely to lie on the motion trajectory.

We note that the construction of initial weights  $\hat{W}$  indicates the likelihood a pose lies on the motion trajectory. Thus, the first active set can be obtained by the initial weights with a threshold  $\varepsilon$ . We use the same threshold strategy to determine the active set at each iteration. Suppose  $A^{(i)} = \{\theta_1, \dots, \theta_p, \theta_{p+1}, \dots, \theta_{p+q}\}$  is the active set at iteration  $i$ , and the corresponding weights  $W_A^{(i)} = \{w_1, \dots, w_{p+q}\}$  is sorted in the descending order  $w_1 \geq \dots \geq w_p \geq \varepsilon \geq w_{p+1} \geq \dots \geq w_{p+q}$ . At iteration  $i+1$ , we set the new active set to be  $A^{(i+1)} = \{\theta_1, \dots, \theta_p, \hat{\theta}_{p+1}, \dots, \hat{\theta}_{p+q}\}$  by deleting the smallest  $q$  weights and adding new poses  $\{\hat{\theta}_{p+1}, \dots, \hat{\theta}_{p+q}\}$  as perturbation. We obtain the new poses by sampling based on the previous active set  $A^{(i)}$  using a Gaussian distribution (with small variance), since the poses close to the members of the active set are most likely to be inliers. The perturbation step makes the proposed algorithm more robust.

Once we determine the active set  $A$ , we assign the weights for the poses of the inactive set to be 0, and estimate the weights of poses in the active set with  $\ell_2$  regularization and non-negative constraints,

$$\min_{W_A} \sum_{\partial_*} \alpha_* \left\| \sum_{\theta \in A} w_\theta \partial_*(K_\theta \tilde{L}) - \partial_* B \right\|^2 + \beta \|W_A\|^2, \quad \text{s.t. } W_A \geq 0. \quad (9)$$

where  $0$  is a zero vector and  $W_A \geq 0$  denotes each weight in  $W_A$  is non-negative. We solve this optimization problem using an interior point method with small  $\beta$  (0.1 in our experiments). Since the active set is already sparse with respect to the whole space, it is not necessary to introduce a sparse regularization term in the above optimization problem.

For parameter settings, we conduct experiments and find that the index number (i.e., the number of elements in a set) of the active set is sensitive when it is smaller than 60 and stable otherwise. The reason is that a very small number of sampled poses may not be enough to cover the motion trajectory so that we could not obtain an accurate weight estimation on such active set. We set the index number to be 60 in this work as it is sufficient to generate good results while the number of samples required to fill the discretized whole motion space is over 1000 even for a small blur kernel size. In [9, 10], since the whole pose space is used for optimization, only coarse-grained sampling is adopted for computational reasons. As we optimize the problem within the active set (which is close to camera motion trajectory), only a small number of poses is required, thereby enabling us to use fine-grained sampling and grids without increasing computational loads. Moreover, it is not necessary to compute all the  $K_\theta$  as a large number of  $K_\theta$  may not be used in the optimization process.

### 4.3 Recovering Latent Image with Deconvolution

In the deconvolution step, we estimate the latent image  $L$  by fixing the weights  $W$ . Since the weights for the poses of the inactive set are set to be 0, we reconstruct the blur matrix with the active set  $A$ ,  $\tilde{K} = \sum_{\theta \in A} w_\theta K_\theta$ , and the optimization problem becomes

$$\min_L \|\tilde{K}L - B\|^2 + \Phi_1(L). \quad (10)$$

We use the regularization terms in [18] and a similar iterative approach to solve this problem.

## 5 Experimental Results

In this section, we evaluate the proposed algorithm against several state-of-the-art single image deblurring methods for spatially variant blur. The experiments are carried out on a machine with 3.40 GHz CPU and 16 GB RAM. Our MATLAB implementation takes about 600 seconds to process a blurry image of  $441 \times 611$  pixels, which is about 1.5 times faster than the MATLAB implementation of [10], the most efficient single image non-uniform deblurring method to the best of our knowledge<sup>1</sup>. Our method with proposed initialization strategy usually takes 5 to 10 iterations to converge with the criteria that the recovered image does not change above a threshold between two iterations. Meanwhile, our method without the initialization strategy takes more than 20 iterations to converge and the results are not satisfying. The source codes and additional results are available at <http://faculty.ucmerced.edu/mhyang/code>.

**Initialization Test using Synthetic Data:** We evaluate the proposed initialization method with synthesized data by manually generating three-dimensional camera motion with a connected path in the motion space and initializing their weights. We then apply the synthesized camera motion on a sharp image to obtain a blurry image, based on geometric model as Eqn. 2. The proposed backprojection method is evaluated using either the ground truth or the estimated kernels as shown in Fig. 2. First, we project the synthesized motion to several image sites and collect the projected blur kernels as the input for backprojection as described in Sec. 4.1. The results in Fig. 2(c) show that the backprojected results are close to the synthesized motion with respect to poses and weights. This experiment demonstrates the backprojection method models the mapping between blur kernels and camera motion well.

<sup>1</sup>We note that the source code of [10] is not available and the time comparison is based on the reported results.

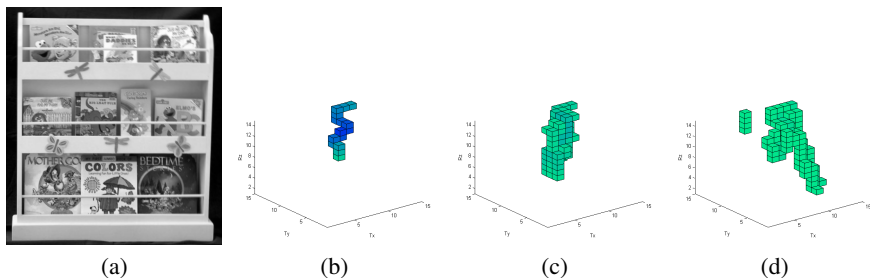


Figure 2: Initialization using backprojection with synthetic data. Light green cubes indicate small values of weight and dark blue ones indicate large values. (a) synthesized blurry image. (b) its corresponding camera motion. (c) initial camera motion using backprojection with ground truth blur kernels at several image sites. (d) initial camera motion using backprojection with blur kernels estimated by a uniform deblurring method [9].

We next evaluate the backprojection method using the estimated blur kernels. Since the kernels of nearby locations are similar due to the spatial continuity, we can assume the blur kernels remain the same within a small region and apply a uniform deblurring algorithm to estimate them. From the blurry image, we select a few image patches from which the blur kernels are estimated using a fast uniform deblurring algorithm [9]. As shown in Fig. 2(d), the estimated poses cover the synthesized motion (i.e., the blocks in (d) covers all blocks in (b)) which indicates that the selected poses are good enough as the first active set for weight estimation. We note the backprojected poses are sparse with respect to the whole sampled pose space (around 100 poses compared with 3375 discretized poses in this case).

**Non-uniform Deblurring:** In Fig. 3, we compare with state-of-the-art single image deblurring algorithms for spatially variant blur [9, 10, 11, 12]. Since not all the source codes are available, we use the reported results for comparisons. Among these methods, three algorithms [9, 10, 12] adopt similar geometric models for camera shake which enforce global constraints for blur kernels while one method [11] estimates kernels locally. In [12], three-dimensional rotations are considered for modeling camera motion while the other methods [9, 10] and our approach focus on estimation of  $x$ ,  $y$  translation and in-plane rotation. We compare mainly with [11] since it achieves the state-of-the-art deblurring results in terms of speed and quality.

For the *Pantheon* image, both the proposed method and [11] recover sharp images without obvious blur while the results from [12] contain ringing effect along the roof. As only three-dimensional rotations are considered in [12], camera translation is not fully removed in this case. In the *Gas station* image, the results from our algorithm contain fewer artifacts and sharper edges (e.g., letters next to the gas pump). The results show that camera motion for this image is better estimated by our approach. In the *Butcher shop* image, most blurs can be removed by [11, 12], although some details are smoothed out when compared with the results generated by the proposed method. We note that an additional step is used in [11] to reduce artifacts using corrective weighting after the latent images are recovered, while our approach does not have any post-processing. However, as shown in the inset images of *Books* and *Butcher shop*, the results from our algorithm recover more details in general. Compared to the results generated by state-of-the-art algorithms, our approach not only removes non-uniform blur efficiently but also generates comparable or better deblurred images.





Figure 3: Comparison with state-of-the-art single image deblurring methods for spatially variant blur. Top to bottom: *Pantheon*, *Gas station*, *Books* and *Butcher shop*.

The main limitation of our approach is that it may fail or require numerous iterations to converge if the blurs are too large (e.g.,  $z$  rotation of 10 degrees), which is also the case for other non-uniform deblurring algorithms [9, 10]. The reason is that blurs resulting from large rotational motion cannot be well estimated from image patches using any uniform deblurring method. Therefore the pose weights are not initialized well, and this likely results in local minimum.

## 6 Conclusions

In this paper we present a fast single image deblurring approach to remove non-uniform blur. We propose an initialization method of camera motion using backprojection to facilitate the convergence and avoid local minimums. We develop a method to solve pose weights and the latent image iteratively. To estimate the pose weights, we restrict the solution space in a small subset for efficiency and add perturbation to make the system robust. Experimental validations with state-of-the-art methods demonstrate the efficiency and effectiveness of our approach. Our future work will focus on developing a public code library of state-of-the-art algorithms with a benchmark dataset and sound metrics for quantitative evaluations.

## References

- [1] J. Bardsley, S. Jefferies, J. Nagy, and R. Plemmons. A computational method for the restoration of images with an unknown, spatially-varying blur. *Optics Express*, 14(5): 1767–1782, 2006.
- [2] J. Cai, H. Ji, C. Liu, and Z. Shen. Blind motion deblurring from a single image using sparse approximation. In *CVPR*, pages 104–111, 2009.
- [3] S. Cho and S. Lee. Fast motion deblurring. In *SIGGRAPH Asia*, 2009.
- [4] S. Cho., Y. Matsushita, and S. Lee. Removing non-uniform motion blur from images. In *ICCV*, 2007.
- [5] T. S. Cho, N. Joshi, C. L. Zitnick, S. B. Kang, R. Szeliski, and W. T. Freeman. A content-aware image prior. In *CVPR*, pages 169–176, 2010.
- [6] T. S. Cho, S. Paris, B. K. P. Horn, and W. T. Freeman. Blur kernel estimation using the radon transform. In *CVPR*, pages 241–248, 2011.
- [7] R. Fergus, B. Singh, A. Hertzmann, S. T. Rowels, and W. T. Freeman. Removing camera shake from a single photograph. In *SIGGRAPH*, pages 787–794, 2006.
- [8] R. Gonzalez and R. Woods. *Digital Image Processing*. Prentice Hall, 2008.
- [9] A. Gupta, N. Joshi, L. Zitnick, M. Cohen, and B. Curless. Single image deblurring using motion density functions. In *ECCV*, pages 171–184, 2010.
- [10] S. Harmeling, M. Hirsch, and B. Scholkopf. Space-variant single-image blind deconvolution for removing camera shake. *NIPS*, 2010.
- [11] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Scholkopf. Fast removal of non-uniform camera shake. In *ICCV*, pages 463–470, 2011.

- 
- [12] J. Jia. Single image motion deblurring using transparency. In *CVPR*, 2007.
  - [13] N. Joshi, S. B. Kang, C. L. Zitnick, and R. Szeliski. Image deblurring using inertial measurement sensors. In *SIGGRAPH*, page 30, 2010.
  - [14] A. Levin. Blind motion deblurring using image statistics. In *NIPS*, pages 841–848, 2006.
  - [15] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *CVPR*, pages 1964–1971, 2009.
  - [16] R. Raskar, A. Agrawal, and J. Tumblin. Coded exposure photography: motion deblurring using fluttered shutter. In *SIGGRAPH*, pages 795–804, 2006.
  - [17] Q. Shan, W. Xiong, and J. Jia. Rotational motion deblurring of a rigid object from a single image. In *ICCV*, 2007.
  - [18] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. In *SIGGRAPH*, pages 73:1–73:10, 2008.
  - [19] Y. Tai, P. Tan, and M. S. Brown. Richardson-lucy deblurring for scenes under projective motion path. *PAMI*, 33(8):1603–1618, 2011.
  - [20] Y. W. Tai, H. Du, M. S. Brown, and S. Lin. Image/video deblurring using a hybrid camera. In *CVPR*, 2008.
  - [21] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. In *CVPR*, pages 491–498, 2010.
  - [22] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *ECCV*, pages 157–170, 2010.