

# Blind restoration of aerial imagery degraded by spatially varying motion blur

Abhijith Punnappurath<sup>a</sup>, A. N. Rajagopalan<sup>a</sup> and Guna Seetharaman<sup>b</sup>

<sup>a</sup>Department of Electrical Engineering, Indian Institute of Technology Madras, Chennai, India

<sup>b</sup>Information Directorate, AFRL/RIEA, Rome NY, USA

## ABSTRACT

This paper deals with deblurring of aerial imagery and develops a methodology for blind restoration of spatially varying blur induced by camera motion caused by instabilities of the moving platform. This is a topic of significant relevance with a potential impact on image analysis, characterization and exploitation. A sharp image is beneficial not only from the perspective of visual appeal but also because it forms the basis for applications such as moving object tracking, change detection, and robust feature extraction. In the presence of general camera motion, the apparent motion of scene points in the image will vary at different locations resulting in space-variant blurring. However, due to the large distances involved in aerial imaging, we show that the blurred image of the ground plane can be expressed as a weighted average of geometrically warped instances of the original focused but unknown image. The weight corresponding to each warp denotes the fraction of the total exposure duration the camera spent in that pose. Given a single motion blurred aerial observation, we propose a scheme to estimate the original focused image affected by arbitrarily-shaped blur kernels. The latent image and its associated warps are estimated by optimizing suitably derived cost functions with judiciously chosen priors within an alternating minimization framework. Several results are given on the challenging VIRAT aerial dataset for validation.

**Keywords:** blind restoration, aerial imagery, spatially varying motion blur, alternating minimization, VIRAT

## 1. INTRODUCTION

Blur in images resulting from motion of the camera during exposure time is an issue in many areas of optical imaging such as remote sensing, aerial reconnaissance and digital photography. For instance, the images captured by cameras attached to airplanes or helicopters are blurred due to both the forward motion of the aircraft, and vibrations. Manufacturers of aerial imaging systems employ compensation mechanisms such as gyroscope gimbals to mitigate the effect of vibrations. Although this reduces the blur due to jitter to some extent, there is no straightforward way to do the same for the forward movement. Moreover, these hardware solutions come at the expense of higher cost, weight and energy consumption. A system that can remove the blur by algorithmic post-processing provides an elegant solution to this problem.

Traditionally, image restoration techniques have modeled blurring due to camera shake as a convolution with a single blur kernel.<sup>1-5</sup> However, it is a well-established fact that the convolution model that employs a uniform blur kernel or point spread function (PSF) across the image is not sufficient to model the blurring phenomenon if the motion is not composed merely of in-plane translations. In fact, camera tilts and rotations occur frequently<sup>6</sup> and the blur induced by camera shake is typically non-uniform. This is especially true in the case of aerial imagery where the blur incurred is not just due to the linear motion of the aircraft but also due to vibrations. Approaches to handling non-uniform blur broadly fall into two categories. The first relies on local uniformity of the blur. Based on the assumption that a continuously varying blur can be approximated by a spatially varying combination of localized uniform blurs, Hirsch et al.<sup>7</sup> propose a method to restore non-uniform motion blur by using an efficient filter flow framework. Building on the idea of a motion density function, yet another scheme for space-varying blur has been proposed by Gupta et al.<sup>8</sup> The motion-blurred image is modeled by considering the camera motion to be comprised only of in-plane translations and in-plane rotations. The second and more recent non-uniform deblurring approach uses an elegant global model<sup>6,9,10</sup> in which the blurred image is represented as the weighted average of warped instances of the latent image, for a constant depth scene. The warped instances

---

E-mail: jithuthatswho@gmail.com, raj@ee.iitm.ac.in, guna@ieee.org

can be viewed as the intermediate images observed by the camera during the exposure time when the camera suffers a shake. Tai et al.<sup>11</sup> have proposed a non-blind deblurring scheme based on modifying the Richardson Lucy deconvolution technique for space-variant blur. However, they assume that the blurring function is known *a priori* and does not need to be estimated. Whyte et al.<sup>6,12</sup> propose a non-uniform image restoration technique where the blurring function is represented on a 3D grid corresponding to the three directions of camera rotations. As pointed out in <sup>10</sup>, the main disadvantage of this global geometric model is heavy computational load due to the dense sampling of poses in the high dimensional camera motion space. A common approach to tackle this problem is to adopt a multi-scale approach that involves constructing an image pyramid and using coarse-grained sampling. But this simplification inevitably introduces reconstruction errors.<sup>10</sup> Hu and Yang<sup>10</sup> present a fast non-uniform deblurring technique by using locally estimated blur kernels to restrain the possible camera poses to a low-dimensional subspace. But the kernels themselves need to be input by the user and the final deblurring quality is dependent on the accuracy of the estimated PSFs. An unnatural  $L_0$  sparse representation for uniform and non-uniform deblurring has also been proposed recently by Xu et al.<sup>13</sup> In the hardware-assisted restoration techniques, Joshi et al.<sup>14</sup> attach sensors to the camera to determine the blurring function, while Tai et al.<sup>15</sup> propose a deblurring scheme that uses coded exposure and some simple user interactions to determine the PSF.

In this work, we propose a fully blind single image non-uniform deblurring algorithm suited for aerial imagery that does not require any additional hardware. We reduce computational overhead by approximating the camera motion with a 3D pose space and optimizing only over a subspace of ‘active’ camera poses. This reduction in dimensionality allows us to use dense sampling and our results compare favourably with state-of-the-art deblurring algorithms. In contrast to <sup>10</sup>, our alternating minimization algorithm, which uses a novel camera pose initialization and pose perturbation step, works on the global geometric model and doesn’t require the calculation of blur kernels at various image locations, thereby eliminating the need for user interaction.

## 2. THE MOTION BLUR MODEL

In this section, we review the non-uniform blur model for aerial images. Since the distances involved are quite large, the ground scene can be modeled as being approximately planar. When the motion of the camera is not restricted to in-plane translations, the paths traced by scene points in the image plane will vary across the image resulting in space-variant blur. The convolution model with a single blur kernel does not hold in such a scenario. However, the blurred image can be accurately modeled as the weighted average of warped instances of the latent image using the projective model in,<sup>6,8,10,16</sup> when the scene is planar. In the discrete domain, this can be represented as

$$b(i, j) = \sum_{k \in \mathbf{T}} \omega(k) l(\mathcal{H}_k(i, j)) \quad (1)$$

where  $l(i, j)$  denotes the latent image of the scene,  $b(i, j)$  is the blurred observation, and  $\mathcal{H}_k(i, j)$  denotes the image coordinates when a homography  $\mathcal{H}_k$  is applied to the point  $(i, j)$ . The parameter  $\omega$ , also called the *transformation spread function* (TSF)<sup>16</sup> in the literature, depicts the camera motion, and  $\omega(k)$  denotes the fraction of the total exposure duration for which the camera stayed in the position that caused the transformation  $\mathcal{H}_k$ . Akin to a PSF,  $\sum_{k \in \mathbf{T}} \omega(k) = 1$ . The TSF  $\omega$  is defined on the discrete transformation space  $\mathbf{T}$  which is the set of sampled camera poses. The transformation space is discretized in such a manner that the difference in the displacements of a point light source due to two different transformations from the discrete set  $\mathbf{T}$  is at least one pixel. Note that although the apparent motion of scene points in the image will vary at different locations when the camera motion is unrestricted, the blurring operation can still be described by a single TSF using equation (1). For example, if the camera undergoes only in-plane rotations, the TSF will have non-zero weights only for the rotational transformations. Observe that if the camera motion is confined to 2D translations, the PSF and TSF will be equivalent.

If  $\mathbf{l}$ ,  $\mathbf{b}$  represent the latent image and the blurred image, respectively, lexicographically ordered as vectors, then, in matrix-vector notation, equation (1) can be expressed as

$$\mathbf{b} = \mathbf{A}\omega \quad (2)$$

where  $\mathbf{A}$  is the matrix whose columns contain projectively transformed copies of  $\mathbf{l}$ , and  $\boldsymbol{\omega}$  denotes the vector of weights  $\omega(k)$ . Note that  $\boldsymbol{\omega}$  is a sparse vector since the blur is typically due to incidental camera shake and only a small fraction of the poses in  $\mathbf{T}$  will have non-zero weights in  $\boldsymbol{\omega}$ . Alternately,  $\mathbf{b}$  can also be represented as

$$\mathbf{b} = \left( \sum_{k \in \mathbf{T}} \omega(k) \mathbf{H}_k \right) \mathbf{l} = \mathbf{B} \mathbf{l} \quad (3)$$

where  $\mathbf{H}_k$  is the matrix that warps the latent image  $\mathbf{l}$  according to the homography  $\mathcal{H}_k$ , while  $\mathbf{B} = \sum_{k \in \mathbf{T}} \omega(k) \mathbf{H}_k$  is the matrix that performs the non-uniform blurring operation. Note that  $\mathbf{B}$  is a sparse square matrix that can be efficiently stored in memory and each row of  $\mathbf{B}$  corresponds to the blur kernel at that particular pixel location.

The homography  $\mathcal{H}_k$  in equation (1) in terms of the camera parameters is given by

$$\mathcal{H}_k = K_v \left( R_k + \frac{1}{d_0} T_k [0 \ 0 \ 1] \right) K_v^{-1} \quad (4)$$

where  $T_k = [T_{X_k} \ T_{Y_k} \ T_{Z_k}]^T$  is the translation vector, and  $d_0$  is the scene depth which is an unknown constant. The rotation matrix  $R_k$  is parameterized<sup>6</sup> in terms of  $\theta_X$ ,  $\theta_Y$  and  $\theta_Z$ , which are the angles of rotation about the three axes. The camera intrinsic matrix  $K_v$  is assumed to be of the form  $K_v = \text{diag}(v, v, 1)$ , where  $v$  is the focal length. Six degrees of freedom arise from  $T_k$  and  $R_k$  (three each). However, it has been shown in<sup>6</sup> that the 6D camera pose space can be approximated by 3D rotations without considering translations when the focal length is large. An alternate approach<sup>8,10</sup> is to model out-of-plane rotations by in-plane-translations under the same assumption of a sufficiently long focal length. This is the approach that we also take in this paper to reduce the dimensionality of the problem i.e., the set of transformations  $\mathbf{T}$  becomes a 3D space defined by the axes  $t_X$ ,  $t_Y$  and  $\theta_Z$  corresponding to in-plane translations along the  $X$  and  $Y$  axes, and in-plane-rotations about the  $Z$  axis, respectively. The homography given in the equation (4) then simplifies to

$$\mathcal{H}_k = \begin{bmatrix} \cos \theta_{Z_k} & -\sin \theta_{Z_k} & t_{X_k} \\ \sin \theta_{Z_k} & \cos \theta_{Z_k} & t_{Y_k} \\ 0 & 0 & 1 \end{bmatrix} \quad (5)$$

where the translation parameters are given by  $t_{X_k} = \frac{v T_{X_k}}{d_0}$  and  $t_{Y_k} = \frac{v T_{Y_k}}{d_0}$ .

### 3. SINGLE IMAGE DEBLURRING

In order to recover the latent image  $l$ , our alternating minimization (AM) algorithm proceeds by updating the estimate of the TSF at one step, and the latent image at the next. We minimize the following energy function over the variables  $\mathbf{l}$  and  $\boldsymbol{\omega}$ ,

$$E(\mathbf{l}, \boldsymbol{\omega}) = \left\| \left( \sum_{k \in \mathbf{T}} \omega(k) \mathbf{H}_k \right) \mathbf{l} - \mathbf{b} \right\|_2^2 + \alpha \Phi_1(\mathbf{l}) + \beta \Phi_2(\boldsymbol{\omega}). \quad (6)$$

The energy function consists of three terms. The first measures the fidelity to the data and emanates from our acquisition model (3). The remaining two are regularization terms on the latent image  $\mathbf{l}$  and the weights  $\boldsymbol{\omega}$ , respectively, with positive weighting constants  $\alpha$  and  $\beta$  that attract the minimum of  $E$  to an admissible set of solutions. The regularization terms will be explained in the following sub-sections. An overview of the proposed algorithm is shown in Fig. 1.

#### 3.1 Initial TSF

Our algorithm requires the user to specify a rough guess of the extent of the blur (translation in pixels along  $X$ ,  $Y$  axes and rotation in degrees about  $Z$  axis) to build the initial TSF. The 3D camera pose space, whose limits are specified by the user, is uniformly sampled to select the initial set of camera poses. We denote this sampled pose space by  $\mathbf{S}$  where  $\mathbf{S} \subset \mathbf{T}$ . In our experiments, the initial TSF contained 200 poses which is still much smaller than the 1500-2000 poses that the whole space  $\mathbf{T}$  would contain even for small to moderate blurs.

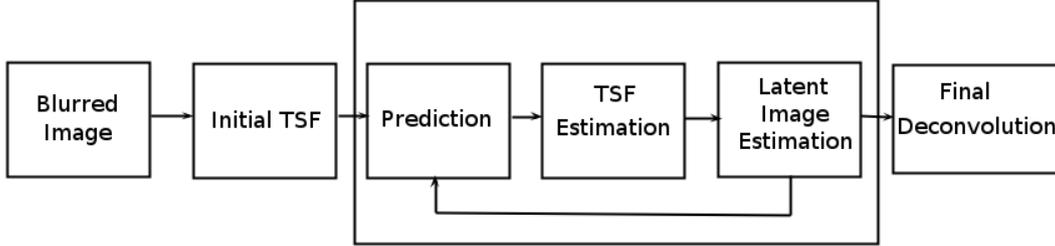


Figure 1. Overview of the proposed deblurring algorithm.

Note that our algorithm requires no other user input. In contrast, Hu and Yang,<sup>10</sup> whose work comes closest to ours, requires the user to input the blur kernels at various locations in the image and we observed that the final deblurring quality depends greatly on the number, location and correctness of these blur kernels. Furthermore, since we model our blur using in-plane rotations and translations, we do not need to know the focal length of the camera as in the case of <sup>6</sup> whose camera pose space is composed of 3D rotations.

### 3.2 TSF Estimation

In the TSF estimation step, we compute  $\omega$  given the current estimate of the latent image  $\mathbf{I}$  based on equation (6).

#### 3.2.1 Image Prediction

Similar to <sup>4</sup>, we perform an image prediction step at each iteration before TSF estimation to obtain more accurate results and to facilitate faster convergence. The prediction step consists of bilateral filtering, shock filtering and gradient magnitude thresholding. Details of the implementation can be found in <sup>4</sup>. The predicted image, denoted by  $\hat{\mathbf{I}}$ , is sharper than the current estimate of the latent image  $\mathbf{I}$  and has fewer artifacts.

#### 3.2.2 TSF estimation on a subspace of $\mathbf{T}$

In the first iteration, we optimize over the initial TSF (obtained in 3.1) by minimizing the following energy function

$$E(\omega) = \|\mathbf{A}\omega - \mathbf{b}\|_2^2 + \beta\Phi_2(\omega) \quad (7)$$

where  $\mathbf{A} = \sum_{k \in \mathbf{S}} \mathbf{H}_k \hat{\mathbf{I}}$  and  $\Phi_2(\omega) = \|\omega\|_1$ . Similar to <sup>4</sup>, we work on gradients instead of image intensities in our implementation of equation (7) since image derivatives have been shown to be effective in reducing ringing effects.<sup>2</sup> This optimization problem can be solved using the nnLeastR function of the Lasso algorithm<sup>17</sup> which considers the additional  $l1$ -norm constraint and imposes non-negativity on the TSF weights. Only the ‘dominant’ poses in the initial TSF  $\mathbf{S}$  are selected as a result of the sparsity constraint imposed by the  $l1$ -norm and the remaining poses which are outliers are removed. We now rebuild the set  $\mathbf{S}$  for the second iteration so that its cardinality is the same as the initial TSF. The new poses are picked around the selected dominant poses by sampling using a Gaussian distribution. This pose perturbation step is based on the notion that the camera trajectory forms a connected 1D path in the camera motion space and, therefore, the poses close to the dominant ones are most likely to be inliers. In the next iteration, equation (7) is minimized over this new ‘active’ set of poses. The variance of the Gaussian distribution is gradually reduced with iterations as the estimated TSF converges to the true TSF. Experiments on synthetic and real data show that our pose perturbation step lends robustness to the algorithm and it does not get stuck in local minima. Note that the number of columns in  $\mathbf{A}$  equals the cardinality of the set  $\mathbf{S}$  which is much less than the total number of poses in  $\mathbf{T}$ . This allows us to compute the matrix  $\mathbf{A}$  at the highest image resolution without running into memory issues. We used a  $\beta$  value of 0.1 for our experiments.

### 3.3 Latent image estimation

In this step, the latent image  $\mathbf{l}$  is estimated by fixing the TSF weights  $\omega$ . The blurring matrix is constructed using only the poses in the active set since the weights of the poses of the inactive set are zero, i.e.  $\mathbf{B} = \sum_{k \in \mathbf{S}} \omega(k) \mathbf{H}_k$  and the energy function to be minimized takes the form

$$E(\mathbf{l}) = \|\mathbf{B}\mathbf{l} - \mathbf{b}\|_2^2 + \alpha \Phi_1(\mathbf{l}) \quad (8)$$

We use the regularization term  $\Phi_1(\mathbf{l}) = \|\nabla l\|_2^2$  in <sup>4</sup> and a conjugate gradient method to solve this problem.

## 4. EXPERIMENTS

This section consists of two parts. We first evaluate the performance of our algorithm on synthetic data and also compare our results with various state-of-the-art single image deblurring techniques. Following this, we demonstrate the applicability of the proposed method on real images using the challenging VIRAT<sup>18</sup> aerial dataset.

We begin with a synthetic example. A latent image of size  $720 \times 600$  pixels is shown in Fig. 2(a). In order to demonstrate our algorithm’s ability to handle 6D motion using just a 3D TSF, we choose the following 6D TSF space- in-plane translations in pixels:  $[-8 : 1 : 8]$ , in-plane rotations:  $[-1.5^\circ : 0.5^\circ : 1.5^\circ]$ , out-of-plane translations:  $[0.95 : 0.05 : 1.05]$  on the image plane, and out-of-plane rotations:  $[\frac{-4}{3}^\circ : \frac{1}{3}^\circ : \frac{4}{3}^\circ]$ . To simulate the motion of the camera, we manually generate 6D camera motion with a connected path in the motion space and initialize the weights. The pose weights in the TSF are defined in such a way that it depicts the path traversed by a camera with non-uniform velocity. The camera motion thus synthesized is applied (using the TSF model) on Fig. 2(a) to produce the blurred image in Fig. 2(b). The results obtained using the deblurring techniques in <sup>13</sup>, <sup>10</sup> and <sup>6</sup> are shown in Figs. 2(c),(d) and (e), respectively. To obtain the result in Fig. 2(f), the following 3D search intervals were input to our algorithm: in-plane translations in pixels:  $[-12 : 1 : 12]$ , in-plane rotations:  $[-2.5^\circ : 0.25^\circ : 2.5^\circ]$ . Our algorithm usually converges within 5 to 8 iterations under the criteria that the recovered image does not change above a threshold between two successive iterations. Note that our result is sharp, free from artifacts and compares closely to the original while the comparison methods have some residual blur. Quantitative comparisons are also provided in Table 1. It can be seen that the proposed method has higher PSNR and SSIM values as compared to the state-of-the-art. Our algorithm and Hu and Yang<sup>10</sup> take approximately 25 minutes on a 3.4 GHz processor running Matlab. The code of Whyte et al.<sup>6</sup> takes much longer in comparison. Although the Matlab implementation of Xu et al.<sup>13</sup> (approximately 15 minutes) runs faster, the output is not satisfactory.

	Xu et al. <sup>13</sup>	Hu and Yang <sup>10</sup>	Whyte et al. <sup>6</sup>	Proposed method
PSNR (in dB)	20.47	19.50	21.68	<b>25.25</b>
SSIM <sup>19</sup>	0.550	0.470	0.597	<b>0.725</b>

Table 1. Comparison with state-of-the-art methods for the synthetic example in Fig. 2.

Next, we test our method on the publicly available VIRAT aerial dataset. Since the database contains videos, we manually extracted some frames to run our algorithm. The frames, shown in the first column of Fig. 3, are at the resolution of the original video i.e.,  $720 \times 480$  pixels. The results obtained using the proposed technique (shown in the second column) clearly demonstrate our algorithm’s ability to produce excellent deblurring results even on real data.

## 5. CONCLUSIONS

This paper described a methodology for blind restoration of aerial images degraded by blur due to camera motion. Due to the large distances involved, we showed that the space-variant blurred image of the ground plane can be expressed as a weighted average of geometrically warped instances of the original image. Given a single observation, the latent image and its associated warps were estimated within an alternating minimization framework. The estimated camera motion itself can be potentially exploited as a valuable cue for stabilization.

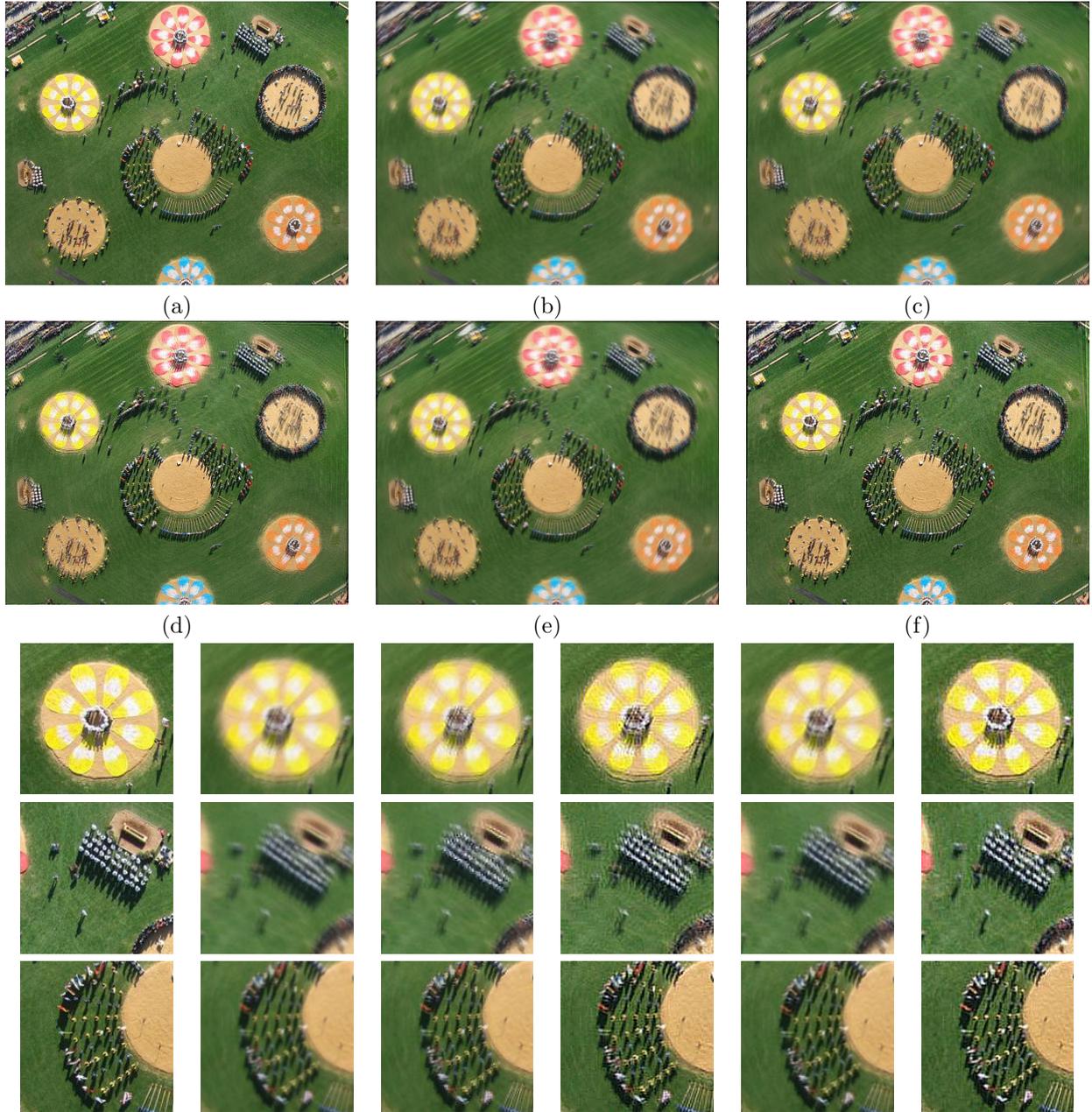


Figure 2. A synthetic example. Rows 1 and 2: (a) Latent image, (b) synthetically blurred image, (c), (d), (e) deblurred outputs obtained using the state-of-the-art deconvolution methods in <sup>13</sup>, <sup>10</sup>, <sup>6</sup>, respectively, (f) deblurred output obtained using the proposed method. Rows 3,4 and 5: Three zoomed-in patches from the images (a) to (f) demonstrating our algorithm’s ability to produce artifact-free deblurred outputs.

Several results were given on synthetic data as well as the challenging VIRAT aerial database for purpose of validation.

### ACKNOWLEDGMENTS

A part of this work was supported by a grant from the Asian Office of Aerospace Research and Development, AOARD/AFOSR. The support is gratefully acknowledged. The results and interpretations presented in this



Figure 3. Deblurring results on the VIRAT aerial database using the proposed method. The first column contains the blurred frames while the second shows deblurred outputs obtained by our algorithm.

paper are that of the authors, and do not necessarily reflect the views or priorities of the sponsor, or the US Air Force Research Laboratory.

## REFERENCES

- [1] Fergus, R., Singh, B., Hertzmann, A., Roweis, S. T., and Freeman, W. T., “Removing camera shake from a single photograph,” in [*ACM Trans. Graph.*], 787–794 (2006).
- [2] Shan, Q., Jia, J., and Agarwala, A., “High-quality motion deblurring from a single image,” *ACM Trans. Graph.* **27**, 73:1–73:10 (Aug. 2008).
- [3] Xu, L. and Jia, J., “Two-phase kernel estimation for robust motion deblurring,” in [*Proc. ECCV*], **6311**, 157–170, Springer (2010).
- [4] Cho, S. and Lee, S., “Fast motion deblurring,” *ACM Trans. Graph. (SIGGRAPH ASIA 2009)* **28**(5), article no. 145 (2009).
- [5] Levin, A., Weiss, Y., Durand, F., and Freeman, W. T., “Understanding blind deconvolution algorithms,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **33**, 2354–2367 (Dec. 2011).
- [6] Whyte, O., Sivic, J., Zisserman, A., and Ponce, J., “Non-uniform deblurring for shaken images,” *International Journal of Computer Vision* **98**, 168–186 (June 2012).
- [7] Hirsch, M., Sra, S., Scholkopf, B., and Harmeling, S., “Efficient filter flow for space-variant multiframe blind deconvolution,” *Proc. CVPR*, 607–614 (2010).
- [8] Gupta, A., Joshi, N., Zitnick, C. L., Cohen, M., and Curless, B., “Single image deblurring using motion density functions,” *Proc. ECCV*, 171–184 (2010).
- [9] Paramanand, C. and Rajagopalan, A. N., “Inferring image transformation and structure from motion-blurred images,” in [*Proc. BMVC*], 1–12 (2010).
- [10] Hu, Z. and Yang, M.-H., “Fast non-uniform deblurring using constrained camera pose subspace,” in [*Proc. BMVC*], (2012).
- [11] Tai, Y.-W., Tan, P., and Brown, M., “Richardson-lucy deblurring for scenes under a projective motion path,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **33**(8), 1603–1618 (2011).
- [12] Whyte, O., Sivic, J., Zisserman, A., and Ponce, J., “Non-uniform deblurring for shaken images,” in [*Proc. CVPR*], 491–498, IEEE (2010).
- [13] Xu, L., Zheng, S., and Jia, J., “Unnatural l0 sparse representation for natural image deblurring,” in [*In Proc. CVPR*], (2013).
- [14] Joshi, N., Kang, S. B., Zitnick, C. L., and Szeliski, R., “Image deblurring using inertial measurement sensors,” *ACM Trans. Graph.* **29**, 30:1–30:9 (July 2010).
- [15] Tai, Y., Kong, N., Lin, S., and Shin, S. Y., “Coded exposure imaging for projective motion deblurring,” in [*In Proc. CVPR*], (2010).
- [16] Paramanand, C. and Rajagopalan, A. N., “Non-uniform motion deblurring for bilayer scenes,” in [*In Proc. CVPR*], (2013).
- [17] Liu, J., Ji, S., and Ye, J., *SLEP: Sparse Learning with Efficient Projections*. Arizona State University (2009).
- [18] Oh, S., Hoogs, A., Perera, A. G. A., Cuntoor, N. P., Chen, C.-C., Lee, J. T., Mukherjee, S., Aggarwal, J. K., Lee, H., Davis, L. S., Swears, E., Wang, X., Ji, Q., Reddy, K. K., Shah, M., Vondrick, C., Pirsiavash, H., Ramanan, D., Yuen, J., Torralba, A., Song, B., Fong, A., Chowdhury, A. K. R., and Desai, M., “A large-scale benchmark dataset for event recognition in surveillance video,” in [*Proc. CVPR*], 3153–3160, IEEE (2011).
- [19] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P., “Image quality assessment: From error visibility to structural similarity,” *IEEE Transactions on Image Processing* **13**(4), 600–612 (2004).