Decomposition of Space-Variant Blur in Image Deconvolution

Filip Sroubek, *Member, IEEE,* Jan Kamenicky, and Yue M. Lu, *Senior Member, IEEE*

***Abstract*—Standard convolution as a model of radiometric degradation is in majority of cases inaccurate as the blur varies in space and we are thus required to work with a computationally demanding space-variant model. Space-variant degradation can be approximately decomposed to a set of standard convolutions. We explain in detail the properties of the space-variant degradation operator and show two possible decomposition models and two approximation approaches. Our target application is space-variant image deconvolution, on which we illustrate theoretical differences between these models. We propose a computationally efficient restoration algorithm that belongs to a category of alternating direction methods of multipliers, which consists of four update steps with closed-form solutions. Depending on the used decomposition, two variations of the algorithm exist with distinct properties. We test the effectiveness of the decomposition models under different levels of approximation on synthetic and real examples, and conclude the letter by drawing several practical observations.**

***Index Terms*—space-variant convolution, singular value decomposition, alternating direction method of multipliers**

I. Introduction

THE last two decades have brought significant progress in the development of efficient methods for classical deconvolution and super-resolution problems in both the single and multi-channel (multiple signals or images) scenarios; see [1], [2] and references therein. Most of these methods work with blurs modeled by convolution, which assumes that the properties of blur are the same in the whole image, so called space-invariant (SI) model. SI deconvolution can be understood also as a normalization process for blur-invariant registration as in [3].

In the majority of practical situations, the blur varies in space and standard convolution does not hold. We refer to this generalized scenario as space-variant (SV) blur. Many attempts were made to estimate SV blur in constrained scenarios, e.g. out-of-focus blur of a 3D scene [4], [5]; blur of moving objects over static background [6]; motion blur induced by camera motion [7], [8]; and experimental measurements of chromatic aberrations [9], [10], just to name few.

Unlike SI degradation (classical convolution), which can be efficiently computed in the Fourier domain, the SV blur can not be simplified in general and this is the main computational bottleneck of any SV deconvolution algorithm. Several methods have been proposed to improve efficiency. The common idea underlying all these methods is to use the fact that a set of blur kernels can be represented by a small number of basis (decomposition) filters. Then SV convolution can be calculated efficiently by doing a combination of parallel filtering and point-wise multiplication. Early attempts in [11] or [12] assumed the decomposition filters to be a subset of the original blur kernels. This idea was generalized using the singular value decomposition (SVD) in [13] and later applied to deconvolution in [14], [15].

In this letter, we present a general framework for decomposition and approximation of SV blur that encompasses all previously proposed approaches. Based on this framework, we present a novel and efficient SV deconvolution algorithm using the alternating direction method of multipliers (ADMM). We also uncover striking differences among various approaches that become visible in the case of fast changing blurs.

The rest of the letter is organized as follows. Sec. II describes the decomposition and approximation models. Two variations of an efficient algorithm are presented in Sec. III. We then demonstrate the effectiveness and efficiency of the proposed method in Sec. IV through experimental results on synthetic and real examples.

II. Decomposition of space-variant blur

Let us consider a model of a static scene with no occlusion under general camera degradation, such as any combination of camera motion, wrong focus, geometric aberration, etc. If we take a particular point in the scene, the light coming from this point is not captured as a point on the sensor at some position s but due to degradation as a point-spread-function (PSF) $h(x)$. The PSF is generally different for each point in the scene and therefore it is also a function of the position s, i.e. $h(x, s)$. Then the SV degradation model can be written in a form naturally generalizing standard convolution as

$$g(x) = \int u(s)h(x - s, s)ds\,, \tag{1}$$

where u is the original image of the scene we want to reconstruct and g is the image captured by the sensor. In the discrete setting, which we shall use from now on, this model can be written in the vector-matrix notation (images are column-wise concatenated) as

$$g = Hu\,, \tag{2}$$

where H is the degradation matrix containing discrete SV kernels $h(x, s)$. Discretized positions x_i and s_j correspond

F. Sroubek and J. Kamenicky are with the Institute of Information Theory and Automation, Czech Academy of Sciences, Prague, Czech Republic. Y. M. Lu is with the John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, USA.

This work was supported by GACR grant GA13-29225S, U.S. NSF grant CCF-1319140, CAS travel grant M100751201 and partially by ARTEMIS JU project 621439 (ALMARVI).

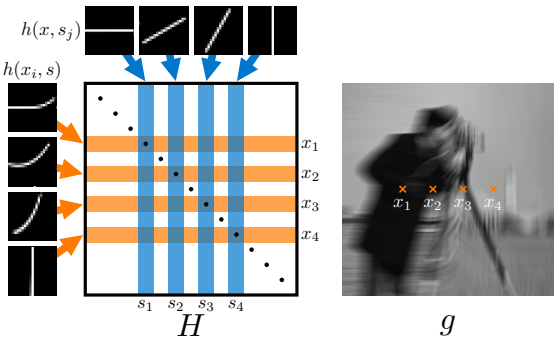


Fig. 1. Illustration of the space-variant degradation operator H. The right image g exhibits motion blur gradually changing from a horizontal to a vertical line. The corresponding degradation matrix H is depicted on left together with the images of blur h that lie in four columns $s_1, \dots, s_4$ and the equivalent four rows $x_1, \dots, x_4$. Note that H is applied to a vectorized image and to obtain the images of h one has to reshape the columns and rows into matrices.

to matrix rows and columns, respectively. For the image size $n \times n$ the length of u and g is $N = n^2$. The matrix H is a block band matrix with maximum of P non-zero entries in every column, where P is the largest number of non-zero pixels $h(x, s)$ from all s. To better illustrate the structure of H, let us consider an example in Fig. 1, in which simple motion blur gradually changes from a horizontal to a vertical line. As follows from (1), each column s_j is populated with the corresponding PSF $h(x, s_j)$ (lines of different orientation shifted to the position s_j). Vectors h_j's of length P will denote these discrete PSFs. Columns define how the corresponding pixels of u are "spread" in the output image g. On the other hand, each row x_i contains elements from different PSFs $\forall s$ $h(x_i - s, s)$. We will refer to row elements as filters f_i. Rows define how the pixels of u are averaged to generate a pixel in g. It is important to understand that the row elements are thus different from the column ones and in our illustrative example they look as curvy lines. Only in the SI case, filters in rows are equal to PSFs in columns rotated by $180°$.

The aim of decomposition is to replace the SV H by a simpler form that allows efficient calculation. The common idea used by all previously published methods is to approximate every local PSF in H as a linear combination of some *decomposition filters* $[b_1, \dots, b_K]$, where $K \ll N$. In this paper, we present a general framework and distinguish between two approaches, which we call the *column-wise decomposition* (CWD) and the *row-wise decomposition* (RWD) of H, respectively.

In CWD, we assume that the PSF in the jth column (h_j) is a linear combination of $[b_1, \dots, b_K]$ with coefficients $[m_j(1), \dots, m_j(K)]$, i.e. $h_j = m_j(1)b_1 + m_j(2)b_2 \dots + m_j(K)b_K$. Let B_k denote a convolution matrix with b_k and M_k a diagonal $N \times N$ matrix with $[m_1(k) \dots, m_N(k)]$ coefficients on the diagonal. Then the original model in (2) becomes

$$g = [B_1, \dots, B_K] \begin{bmatrix} M_1 \\ \vdots \\ M_K \end{bmatrix} u = BMu, \qquad (3)$$

where $B = [B_1, \dots, B_K]$ is a block convolution matrix of size $N \times KN$ and $M = [M_1^T, \dots, M_K^T]^T$ is a block diagonal matrix of size $KN \times N$. The RWD, on the other hand, decomposes rows instead of columns. Then the filter on the ith row (f_i) is a linear combination of $[b_1, \dots, b_K]$ with coefficients $[m_i(1), \dots, m_i(K)]$ and

$$g = [M_1, \dots, M_K] \begin{bmatrix} B_1 \\ \vdots \\ B_K \end{bmatrix} u = M'B'u\,. \qquad (4)$$

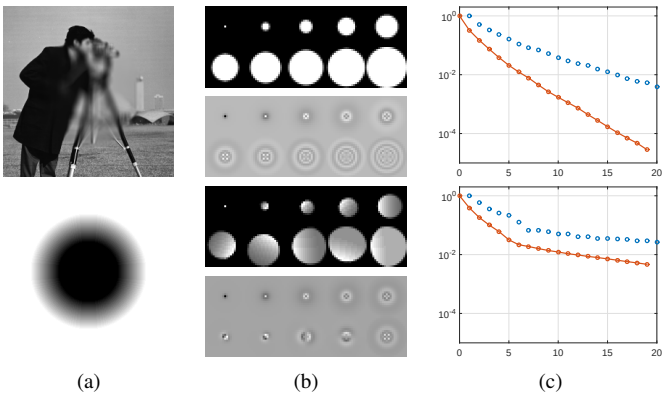
Comparing (3) and (4), we can see that the order of convolution and diagonal matrices is swapped. Note that the block matrices M' and B' have different shape and that the decomposition filters and coefficients are generally different from CWD.

The next critical step is to determine decomposition filters b_k's. Note that there is no specific constraint on the decomposition filters such as e.g. orthogonality. The only property that we seek is to approximate well the original PSFs with small K since then the deconvolution method will be efficient. Two approaches exist in the literature: direct use of original PSFs and SVD decomposition.

The first one, as proposed originally in [11], uses a selection of the original PSFs that lie on a sparse spatial grid and expresses PSFs lying in between as a bilinear interpolation of the PSFs on the grid. Then for CWD, the decomposition filters b_k's are simply a subset of the PSFs in columns ($b_k \in \{h_j\}_1^N$) and $m_j(k)$ are the bilinear interpolation coefficients if the grid is rectilinear. In this case, each column of M contains at most 4 non-zero elements, since bilinear interpolation uses only 4 closest b_k's. This was for example implemented in [12]. For an uneven grid one can apply triangulation followed by barycentric interpolation but this has not been considered in the literature to our knowledge. The original work of Nagy *et al.* [11] implemented RWD but instead of using $b_k \in \{f_i\}_1^N$ the original PSFs in columns h_j's were used, which is less optimal as we show later.

The second approach, proposed independently in [14] and [15], takes a different path and estimates b_k's by applying SVD to the original PSFs. Miraut *et al.* [14] implemented CWD and SVD was applied to $\{h_j\}_1^N$. Deng *et al.* [15] adopted RWD but the decomposition filters were calculated also from $\{h_j\}_1^N$, which is not optimal. Here we sketch the SVD approach for CWD. (The RWD derivation is the same, except that f_i's are used instead of h_j's.) Let us arrange h_j in a matrix $A = [h_1, \dots, h_N]$ of size $P \times N$, $P < N$. SVD decomposes A to $A = USV^T$. Keeping the largest K singular values $S_K \equiv \mathrm{diag}(\{\sigma_1, \dots, \sigma_K\})$, the K-rank approximation of A is a truncated matrix $A_K = US_KV^T$ with an approximation MSE $\epsilon(K) = \|A - A_K\|^2 = \sum_{i=K+1}^{N} \sigma_i^2$. If A is compressible (i.e., of approximately low rank), the singular values decrease quickly and so does ϵ. Then a relatively small K provides a good approximation of A and the first K columns of U are our decomposition filters $[b_1, \dots, b_K]$ and $m_j = [\sigma_1 V(j,1), \dots, \sigma_K V(j,K)]^T$.

Let us demonstrate the SVD approximation and the difference between RWD and CWD on a simple example of



(a) (b) (c)

Fig. 2. Cylinder blur: column-wise approximation on top and row-wise approximation on bottom; (a) space-variant blur with cylinder radius changing according to the mask on bottom, (b) original blurs and corresponding decomposition filters, (c) approximation error.

cylinder blur with a SV diameter in Fig. 2. This type of blur simulates out-of-focus blur and variations in the blur size are due to changing scene depth. The diameter varies from 1 to 21 pixels according to the mask [(a) bottom] and this results in the image [(a) top]. The top two figures in (b) correspond to CWD; examples of 10 PSFs in columns (identical to the cylinder blurs) and the first 10 decomposition filters ordered according to their singular values. Unlike the PSFs, the decomposition filters due to construction must be orthogonal and can have negative values and therefore the gray color in the background corresponds to zero. The bottom two figures in (b) correspond to the RWD. We see that the filters in rows are different from the PSFs and the decomposition filters also differ. It is interesting to observe in (c) how the approximation error decreases with increasing numbers of K. The semi-log plots show relative values, i.e. σ_i/σ_1 (circles) and $\epsilon(i)/\epsilon(0)$ (connected circles), where $\epsilon(0) = \|A\|^2$. The top plot is for CWD and the bottom plot is for RWD. The MSE decreases faster in CWD and this is also true in other SV scenarios we tested in the supplementary material.

There are two theoretical advantages of using CWD over RWD. Whether we use bilinear interpolation or the SVD for approximation, the first advantage is that we work with h_j's, which are directly measurable since they correspond to observations of point sources (PSFs). On the other hand, RWD requires an additional step of constructing blocks of the matrix H to calculate f_i's, which are neither directly observable nor can be estimated via blind deconvolution methods. In the case of the SVD approximation, the second advantage is a faster decrease of MSE and thus potentially smaller K necessary for artifact-free deconvolution.

III. Algorithm

Having the decomposition of H, we now propose two algorithms for CWD and RWD, respectively, that estimate the original image u (non-blind SV deconvolution). Let us start with the CWD in (3) and include noise n, which gives us an acquisition model

$$g = BMu + n\,. \tag{5}$$

We estimate u by solving a regularized least-squares problem (or equivalently, computing the maximum a posteriori estimator)

$$\min_u \frac{\mu}{2}\left\|BMu - g\right\|_2^2 + \alpha\left\|Cu\right\|_1, \tag{6}$$

where the first term is the data fidelity term coming from the model (5) and the second term ensures image regularization. Generally, C is a linear analysis operator (e.g. the gradient operator or the forward wavelets transform), which converts the image into a sparse domain.

To solve the problem (6) efficiently, the state-of-the-art approach is to use ADMM, which is equivalent to variable splitting with the augmented Lagrangian method (ALM). We propose two variable splittings that transform the original problem to a constraint $\min_{u,v,w} \frac{\mu}{2}\left\|Bw - g\right\|_2^2 + \alpha\left\|v\right\|_1$, s.t. $v = Cu$, $w = Mu$. The constrained minimization is converted to an unconstrained one by ALM, which yields the final alternating minimization problem

$$\begin{aligned}\min_{u,v,w} &\frac{\mu}{2}\left\|Bw - g\right\|_2^2 + \alpha\left\|v\right\|_1 \\ &+ \frac{\gamma}{2}\left\|Mu - w - q\right\|_2^2 + \frac{\beta}{2}\left\|Cu - v - p\right\|_2^2.\end{aligned} \tag{7}$$

Variables p and q are introduced by ALM and they are proportional to the Lagrangians of the equality constraints.

Minimization with respect to u requires an inversion of $\gamma M^TM + \beta C^TC$. Operator C is chosen such that C^TC (e.g. Laplacian operator) is easily calculated in the Fourier domain. However, the first term (diagonal matrix) cannot be expressed in the frequency spectrum and therefore the inversion cannot be computed efficiently. To overcome this problem, we propose to modify the third term in (7) to $\frac{\gamma}{2}\left\|u - M^+(w+q)\right\|_2^2$, where $M^+ = (M^TM)^{-1}M^T$ is the pseudo-inverse of M. Now we have to invert $\gamma I + \beta C^TC$ and this time it can be easily done in the Fourier domain.

The alternating minimization in (7) is done by iteratively solving four update steps:

$$\begin{aligned}
&1)\ \min_u \Rightarrow \ \left(\gamma I + \beta C^TC\right)u = \gamma M^+(w+q) + \beta C^T(v+p)\\
&2)\ \min_w \Rightarrow \ \left(\mu B^TB + \gamma I\right)w = \mu B^Tg + \gamma(Mu - q)\\
&3)\ \min_v \Rightarrow \ v = \frac{Cu-p}{|Cu-p|}\max\left(|Cu-p| - \frac{\alpha}{\beta}, 0\right)\\
&4)\quad q^{i+1} = q^i - Mu + w, \quad p^{i+1} = p^i - Cu + v
\end{aligned}$$

The index i denotes iterations but for the sake of readability the index was omitted in u, v and w. Note, that all steps lead to explicit solution. The first two can be efficiently computed in the Fourier domain, the latter two are simple per-pixel equations.

The derivation of the ADMM algorithm for RWD is similar, but the swapped order of operators B' and M' requires slightly modified variable splitting, $v = Cu$ and $w = B'u$. Update steps 3 and 4 are identical except M is replaced by B' in the update equation for q. The first two steps are:

$$\begin{aligned}
&1)\qquad \left(\gamma B'^TB' + \beta C^TC\right)u = \gamma B'^T(w+q) + \beta C^T(v+p)\\
&2)\qquad \left(\mu M'^TM' + \gamma I\right)w = \mu M'^Tg + \gamma(B'u - q)
\end{aligned}$$

Note that in both CWD and RWD, step 1 are N per-pixel equations and step 2 are N $K \times K$ linear equations but inversions can be precomputed once at the beginning.

Theoretical advantages of CWD were discussed earlier but the RWD algorithm has one important practical advantage. The remedy for convolution boundary effects by so-called "masking" proposed in [16] can be seamlessly incorporated by zeroing boundary pixels in M'. This can not be done in the CWD algorithm due to swapped order of M and B.

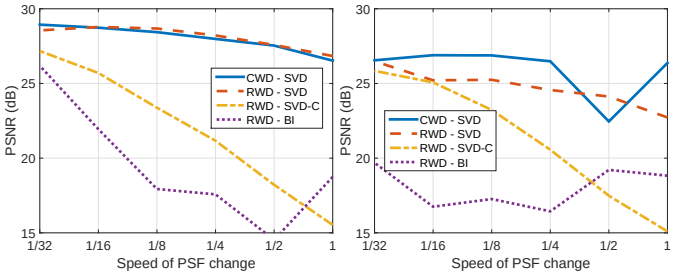
IV. Experiments and Discussion

The experiment in Figs. 3 and 4 summarizes our comparison of different decomposition and approximation approaches of SV out-of-focus blur (cylinder PSF). The experimental set-up simulates capturing a slanted wall. The radius of the cylinder PSF increases in the image horizontally from 1 to 10 pxs with a speed depending on the wall angle. The maximum speed (no occlusion), which we set equal to 1, is the radius increase of 1 px every 1 column in the image; see example in Fig. 3(a). Four different combinations of decomposition and approximation were tested: CWD with SVD approximation of columns h_j's (CWD-SVD), RWD with SVD approximation of rows f_i's (RWD-SVD), RWD with SVD approximation of columns h_j's (RWD-SVD-C), RWD with approximation by bilinear interpolation (RWD-BI). In all four cases, the approximation was done with $K = 10, 20$ and results were averaged over 10 different images (nature, faces, buildings, signs, etc.) generating in total of eight plots of deconvolution PSNR versus speed in Fig. 4. In the case of $K = 20$ the performance of all four methods is more even with respect to speed than for $K = 10$, where the approximation errors start to dominate. Compare visually reconstruction artifacts in Fig. 3(b)-(d) for speed $1/2$, where the top row shows $K = 20$ and the bottom row $K = 10$. The best results are achieved with the CWD-SVD and RWD-SVD, which perform equally well for high K as expected since both approaches correctly approximate the same matrix H. For small K, the CWD-SVD slightly outperforms the RWD-SVD, which we relate to the faster decrease of singular values in CWD discussed earlier. It is important to notice that applying SVD on PSFs but using RWD (RWD-SVD-C) or using directly PSFs with bilinear interpolation (RWD-BI or CWD-BI, which is not presented here) as commonly used in the literature gives inferior performance. However, the differences among various decomposition and approximation approaches diminish if the speed of PSF changes is small and sufficiently large K is used. Slowly varying SV scenarios were predominantly considered in the literature and this is the reason we believe, why the differences in decomposition were not observed previously.

The experiment in Fig. 5 illustrates SV deconvolution on a real photo taken by a traffic camera. The observed car shows severe motion blur. The SV nature of the blur is visible by comparing the directions of line blur in different locations; see Fig. 5(a). The epipole of the blur lines was manually extracted and used to calculate the blur direction and length in every pixel of the car front. Fig. 5(b) compares the SI deconvolution (top) with the SV method (bottom) implemented as CWD with

(a) original / input (b) CWD-SVD (c) RWD-SVD-C (d) RWD-BI

Fig. 3. Space-variant deconvolution: (a) ideal sharp image and input blurred image, (b) column-wise decomposition with SVD approximation, (c) row-wise decomposition with SVD from PSFs in columns, (d) row-wise decomposition with bilinear approximation. In (b)-(d), top row is for $K = 20$ decomposition filters and bottom row is for $K = 10$.

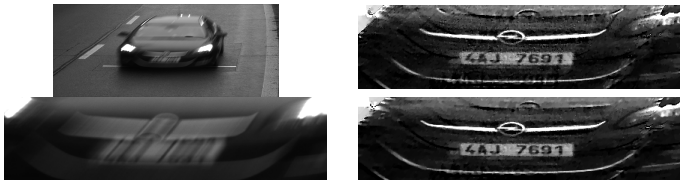


(a) $K = 20$ (b) $K = 10$

Fig. 4. PSNR of the results of individual methods for different speed of PSF changes: (a) number of decomposition filters $K = 20$ and (b) $K = 10$.

SVD approximation. More experiments can be found in the supplementary material and the MATLAB code is available at `http://zoi.utia.cas.cz/decomposition`.

To conclude this letter, we summarize some practical observations. It is important to distinguish between the row and column decomposition and from which set of filters we perform approximation. Considering the convolution boundary effect, which is present in many practical applications, the best choice is to use RWD and correctly apply SVD on row filters. The only drawback is the extra step of estimating row filters from PSFs. If the SV blur variations are slow then we can do equally well with RWD and bilinear interpolation of PSFs, provided that we have sufficiently dense interpolation grid. The proposed method cannot handle occlusion, which requires a more complex model than the SV blur considered here.



(a) (b)

Fig. 5. Real-data experiment: (a) input image and cropped region, (b) space-invariant deconvolution on top and space-variant on bottom. We would like to thank Lukaš Maršík from CAMEA Ltd. for providing the blurred images.

REFERENCES

[1] P. Campisi and K. Egiazarian, Eds., *Blind Image Deconvolution, Theory and Application*, CRC Press, 2007.

[2] P. Milanfar, Ed., *Super-resolution Imaging*, CRC Press, 2010.

[3] M. Pedone, E. Bayro-Corrochano, J. Flusser, and J. Heikkila, "Quaternion Wiener deconvolution for noise robust color image registration," *IEEE Signal Processing Letters*, vol. 22, no. 9, pp. 1278–1282, 2015.

[4] M. Šorel and J. Flusser, "Space-variant restoration of images degraded by camera motion blur," *IEEE Transactions on Image Processing*, vol. 17, no. 2, pp. 105–116, Feb. 2008.

[5] Z. Hu, L. Xu, and M.-H. Yang, "Joint depth estimation and camera shake removal from single blurry image," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, 2014, pp. 2893–2900.

[6] J. Jia, "Single image motion deblurring using transparency," in *Computer Vision and Pattern Recognition (CVPR), 2007 IEEE Conference on*, 2007, pp. 1–8.

[7] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, "Non-uniform deblurring for shaken images," *International Journal of Computer Vision*, vol. 98, no. 2, pp. 168–186, 2012.

[8] O. Šindelář and F. Šroubek, "Image deblurring in smartphone devices using built-in inertial measurement sensors," *Journal of Electronic Imaging*, vol. 22, no. 1, pp. 011003–011003, 2013.

[9] C.J. Schuler, M. Hirsch, S. Harmeling, and B. Scholkopf, "Non-stationary correction of optical aberrations," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, Nov 2011, pp. 659–666.

[10] R. Tezaur, T. Kamata, L. Hong, and S.S. Slonaker, "A system for estimating optics blur psfs from test chart images," in *Proc. SPIE*, 2015, vol. 9404, pp. 94040D–94040D–10.

[11] J.G. Nagy and D.P. O'Leary, "Restoring images degraded by spatially variant blur," *SIAM Journal on Scientific Computing*, vol. 19, no. 4, pp. 1063–1082, 1998.

[12] M. Hirsch, S. Sra, B. Scholkopf, and S. Harmeling, "Efficient filter flow for space-variant multiframe blind deconvolution," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, June 2010, pp. 607 –614.

[13] T. Popkin, A. Cavallaro, and D. Hands, "Accurate and efficient method for smoothly space-variant Gaussian blurring," *IEEE Transactions on Image Processing*, vol. 19, no. 5, pp. 1362–1370, 2010.

[14] D. Miraut and J. Portilla, "Efficient shift-variant image restoration using deformable filtering (part i)," *EURASIP Journal on Advances in Signal Processing*, vol. 2012, no. 1, pp. 100, 2012.

[15] H. Deng, W. Zuo, H. Zhang, and D. Zhang, "An additive convolution model for fast restoration of nonuniform blurred images," *International Journal of Computer Mathematics*, vol. 91, no. 11, pp. 2446–2466, 2014.

[16] A. Matakos, S. Ramani, and J.A. Fessler, "Accelerated edge-preserving image restoration without boundary artifacts," *IEEE Transactions on Image Processing*, vol. 22, no. 5, pp. 2019–2029, May 2013.