Grid Warping Postprocessing for Linear Motion Blur in Images

Andrey Nasonov, Yakov Pchelintsev, Andrey Krylov
Laboratory of Mathematical Methods of Image Processing
Faculty of Computational Mathematics and Cybernetics
Lomonosov Moscow State University
Russia, Moscow, Leninsky gory, 1, BMK MGU
Email: kryl@cs.msu.ru

***Abstract*—The paper presents a method for linear motion blur and out-of-focus blur suppression in photographic images. Conventional image deconvolution algorithms usually have a regularization parameter that specifies a trade-off between incomplete blur removal and high probability of artifacts like ringing and noise. The idea of the proposed image deblurring method is to apply grid warping approach to improve image sharpness after conventional image deconvolution algorithms used with strong regularization. Grid warping algorithm moves pixels in edge neighborhood area towards the edges making them sharper without introducing artifacts like ringing and noise. The proposed method is expected to have the same sharpness but decreased risk of artifacts compared to standalone image deconvolution. In order to validate the proposed scheme, we have applied it to artificially blurred images and images with real blur, with different levels of noise and blur radii, directions and lengths.**

***Index Terms*—edge sharpening, image deblurring, grid warping, optical blur, 3D image sharpening**

I. Introduction

Image blur is a common defect in photographic images. It is caused by camera and object motion during exposure or wrong focusing. during the exposure period, the movements of the camera or the objects produce motion blurred images, as the luminance changes are integrated over time.

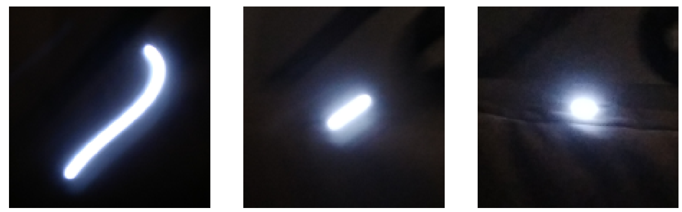
Development of image deblurring algorithms is one of the most important image processing problems. Image deblurring is usually posed as a classical linear inverse problem. It consists of finding an unknown image from blurry and noisy observation:

$$z_B = z * H + n,$$

where z_B is an observed image, H is the blur kernel and n is additive noise. If the blur kernel H and the information about the noise n are known with sufficient precision, the deconvolution problem can be effectively solved by numerous image deconvolution algorithms [1]–[3].

In practice, the blur kernel H and noise level n are not known and their estimation is a challenging problem. Blind deconvolution algorithms can be used to solve this problem [4]–[6] but their results are unpredictable in many real applications due to high complexity and non-uniformity of blur kernel.

One of the ways to reduce blur kernel complexity and to improve the effectiveness of blur kernel estimation algorithms is to reduce the exposure time. The side effect is a proportional increase of the noise level. In order to compensate this effect, several techniques have been developed. One of them can be summarised as taking a series of short exposure frames and programmatically combining them into one image [7]. Another approach is to take blurred-noisy image pair and to use it for the blur kernel estimation [8].



(a) 1/15 s (b) 1/60 s (c) 1/250 s

Fig. 1. Comparison of motion blur kernels at different exposures.

An increase of modern cameras sensor resolution has resulted in visible motion blur even at short exposure time. Compared to long exposure motion blur, a short exposure blur kernel has significantly simpler structure (see Fig. 1). Linear blur model is used to approximate the motion blur at short exposures.

Different linear blur detection and suppression algorithms have been developed recently. The work [9] proposes a deep learning-based approach for estimation of the non-uniform motion blur, followed by a patch statistics-based deblurring model adapted to non-uniform motion blur. The method [10] is based on the spectrum of the blurred images and operates with an assumption which is valid for most natural images: the power-spectrum is approximately isotropic and decreases exponentially for higher frequencies. In [11], the blur-field estimation is also based on the analysis of the spectral content of blurry image patches.

Although the blur kernel has become simpler, deconvolution algorithms can still produce unwanted artifacts like ringing and noise amplification. This risk can be reduced by adjusting the parameters of deconvolution algorithms at the cost of keeping residual blur.

An alternative to classical deconvolution algorithms is grid warping approach. Its idea is to transform the neighborhood of the blurred edge so that the pixels move closer to the edge. Since pixel values do not change, the grid warping approach cannot amplify noise or introduce ringing artifact. The warping approach for image enhancement has been introduced in [12]. In that work pixel grid warping is performed according to the solution of a differential equation derived from the warping process constraints. The solution of the equation is used to move the edge neighborhood pixels towards the edge, and the areas between edges are stretched. In [13] warping is computed directly using the values of left and right derivatives.

The limitation of the grid warping approach is that it is applied only to edges and its use as a standalone algorithm is limited. The works [14], [15] present an improved grid warping algorithm and suggest using grid warping as an effective post-processing algorithm for image deblurring for Gaussian blur.

In our paper, we address the problem of image deblurring with linear motion blur combined with out-of-focus blur and propose a combined image deblurring algorithm that has the advantages of both deconvolution and grid warping approaches. The image is first deblurred by a deconvolution algorithm and then the image edges are further sharpened by grid warping approach.

II. Blur model

We use the following blur model:

$$PSF[r, \sigma, \ell, \theta] = C[r] * L[\ell, \theta] * G[\sigma]. \quad (1)$$

It is a combination of out-of-focus blur $C[r]$, linear motion blur $L[\ell, \theta]$ and other types of blur $G[\sigma]$.

The shape of out-of-focus blur is camera specific. We model it as a circular blur with radius r:

$$C[r](x, y) = \begin{cases} \frac{1}{\pi r^2}, & x^2 + y^2 \leq r, \\ 0, & \text{otherwise.} \end{cases}$$

Linear motion blur with length ℓ and direction ϕ is formulated as:

$$L[\ell, \phi](x, y) = L[\ell, 0](x \cos\phi - y \sin\phi, x \sin\phi + y \cos\phi),$$

$$L[\ell, 0](x, y) = \frac{1}{\ell}\theta(\ell/2 - |x|)\delta(y),$$

where $\delta(t)$ and $\theta(t)$ are Dirac and Heaviside functions respectively:

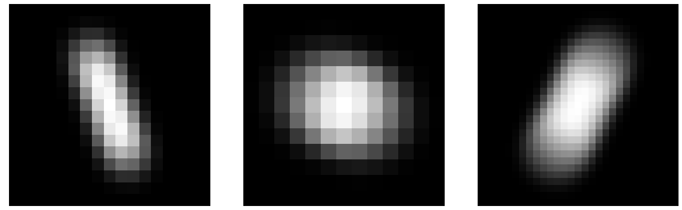
$$\int_{-\infty}^{+\infty} \delta(t) f(t) dt = f(0),$$

$$\theta(t) = \begin{cases} 1, & t \geq 0, \\ 0 & t < 0. \end{cases}$$

We model the remaining blur (for example, an anti-aliasing filter) as a Gaussian blur with parameter σ:

$$G[\sigma](x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right).$$

Some examples of modeled blur are presented in Fig. 2.



(a) $r = 2, \ell = 10$ (b) $r = 3, \ell = 4$ (c) $r = 5, \ell = 16$

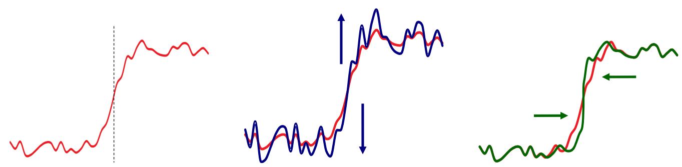
Fig. 2. Different blur kernels according to the model [?].

III. Grid warping

In this section we describe the idea of edge enhancement using pixel grid transformation.

A. One-dimensional edge sharpening

The profile of a blurred edge is more gradual compared to a sharp edge profile. In order to make the edge sharper its transient width should be decreased (see Fig. 3).



(a) Edge profile (b) Typical image enhancement approach (c) Warping approach: pixels are shifted

Fig. 3. The idea of edge sharpening by grid warping.

For any edge profile $g(x)$ centered at $x = 0$ its sharper version $h(x)$ can be obtained shifting the pixels towards the edge center. The *displacement function* $d(x)$ describes the shift of a pixel with a coordinate x to a new coordinate $x + d(x)$ [15]:

$$h(x + d(x)) = g(x).$$

The warped grid should remain monotonic:

$$x_1 < x_2 \Rightarrow x_1 + d(x_1) < x_2 + d(x_2),$$

so the derivative of the displacement function should match the following constraint:

$$d'(x) \geq -1. \quad (2)$$

Another constraint localizes the area of warping effect in the edge neighborhood:

$$d(x) \to 0, \qquad \text{for } |x| \to \infty.$$

The displacement function $d(x)$ greatly influences the result of the edge warping. On the one hand, the edge slope should become steeper. On the other hand, the area near the edge should not be stretched over some predefined limit to avoid wide gaps between adjacent pixels in the discrete case.

The displacement function can be expressed by the proximity function [15]:

$$p(x) = d'(x) + 1.$$

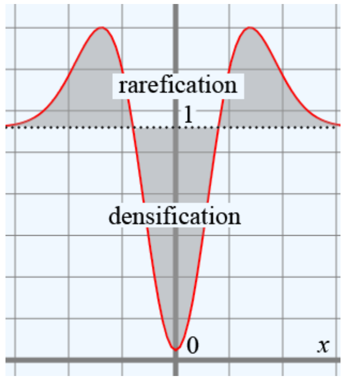


Fig. 4. The shape of the proximity function (3).

The proximity function defines the distance between adjacent pixels after the image warping. This value is inverse to the density value. If the proximity function $p(x)$ is less than 1, then the area is densified at the point x. If the proximity is greater than 1, then the grid is rarefied. For an unwarped image $p(x) \equiv 1$.

We use the proximity function which is a difference between two Gaussian functions [15]:

$$p(x) = 1 - \frac{G[k\sigma_0](x) - G[\sigma_0](x)}{G[k\sigma_0](0) - G[\sigma_0](0)} \tag{3}$$

where σ_0 corresponds to blur level and $k > 1$ controls the size of rarefication area. We use $k = 2$. Fig. 4 demonstrates the shape of the proximity function.

B. Two-dimensional algorithm

In two-dimensional case, the displacement of each pixel is a vector $\vec{d}(x, y)$. Pixels are surrounded by multiple edges and the calculation of displacements is not obvious.

The simplest approach to obtain the displacement for a given pixel is to find the nearest edge, take a one-dimensional section connecting the nearest edge pixel and the given pixel and calculate the displacement of the section using one-dimensional algorithm.

A more effective approach is to use all edges from the pixel neighborhood and perform weighted averaging (see Fig. 5):

$$\vec{d}(P) = \frac{\sum\limits_{Q \in N(P) \cap \{E\}} w(P,Q) \cdot \vec{n}(Q) \cdot d(x(P,Q))}{\sum\limits_{Q \in N(P) \cap \{E\}} w(P,Q)}, \tag{4}$$

where:

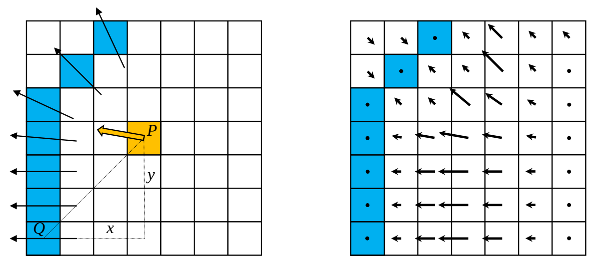
- $d(t)$ is a one-dimensional displacement function;
- $\{E\}$ is the set of edge pixels in the image;
- $N(P)$ is the neighborhood of a point P;
- $\vec{n}(Q) = \frac{\vec{g}(Q)}{|\vec{g}(Q)|}$ is the unit vector corresponding to the edge profile (gradient) direction;
- $x(P, Q)$ is the projection of the vector $\overrightarrow{PQ}$ onto the edge profile;
- $w(P, Q)$ is the weight coefficient.

The size of the neighborhood corresponds to the support range of the displacement function $d(t)$. The weight coefficient $w(P, Q)$ is defined as

$$w(P,Q) = |\vec{g}(Q)| \exp\left(-\frac{y(P,Q)^2}{2\sigma_0^2}\right),$$

where $y(P, Q)$ is the rejection of the vector $\overrightarrow{PQ}$ onto the edge profile direction:

$$y(P,Q)^2 = |\overrightarrow{PQ}|^2 - x(P,Q)^2.$$



(a) The displacement vector for yellow pixel is calculated as a weighted sum of displacement vectors for each of the edge pixels in the neighboorhood of the yellow pixel.

(b) The resulting displacement vector field.

Fig. 5. An illustration for two-dimensional grid warping calculation. Blue pixels belong to edges.

The advantage of the multi-edge algorithm compared to the single-edge algorithm is the improved image quality due to smoother vector field [16]. At the same time, single-edge algorithm is significantly faster.

The multi-edge algorithm also supports handling individual blur level in each pixel. In this case, the warping vectors are calculated as:

$$\vec{d}(P) = \frac{\sum\limits_{Q \in E(P)} w(P,Q) \cdot \vec{n}(Q) \cdot d[Q](x(P,Q))}{\sum\limits_{Q \in E(P)} w(P,Q)}, \tag{5}$$

The only difference between (4) and (5) is choosing individual displacement function $d[Q]$ for each pixel Q.

IV. Experiments

The method has been evaluated on a set of images with modeled linear blur.

A. Scenario

We took 24 reference images from TID database [17] and applied linear blur (1) with random parameters in the following ranges: $\sigma \in [1, 5]$, $\ell \in [1, 16]$, $\theta \in [0, 2\pi]$. We also added Gaussian noise with random variance $\sigma_n \in [2, 32]$. A total of 240 blurred images were generated, 10 samples per each reference image.

We applied a deblurring algorithm followed by edge sharpening using grid warping. The deblurring algorithm and warping were performed for blur parameters $r_B = r * q_B$ and $r_W = r * q_W$ respectively, $q_B, q_W \in (0, 1]$.

The images were deblurred using the state-of-the-art image deconvolution algorithm based on Total Generalized Variation (TGV) regularization [3]:

$$z_R = \arg\min_{v,z} \|Az - u\|_2 + \gamma_1 \|\nabla z - v\|_1 + \gamma_2 \|\nabla v\|_1,$$

where A is the blur kernel, u is the input blurred and noisy image, v is an auxiliary variable, z is the restored image, γ_1 and γ_2 are the regularization parameters that were optimized for each image independently to maximize SSIM value.

In real applications, blur parameters are not given and have to be estimated. In order to model blur kernel estimation errors, we applied a distortion to known blur parameters: the values r and ℓ were multiplied by random values from range $[0.8, 1.2]$, and a random value from range $[-\frac{\sigma\pi}{3\ell}, \frac{\sigma\pi}{3\ell}]$ was added to the angle θ.

B. Results

During the experiment, we calculated average PSNR and SSIM values for different q_B and q_W combinations. The best choice in all cases was to use $q_B = 1$ to suppress the maximum possible blur at the deconvolution stage. At the same time, it can be seen that the deconvolution algorithm was unable to completely suppress all the blur. The remaining blur was effectively reduced by the grid warping algorithm.

Some examples of the obtained results are shown in Fig. 6 and Fig. 7.

Fig. 8 shows the dependence of PSNR and SSIM on q_B and q_W values for difference noise levels.

In can be seen that the warping algorithm improves the result of Total Generalized Variation algorithm. The increase of PSNR and SSIM metrics may look insignificant. This is caused by the fact that grid warping is applied only to small area of the image containing edges and its contribution to overall metric values is low. Nevertheless, the improvement of edge sharpness is clearly visible.

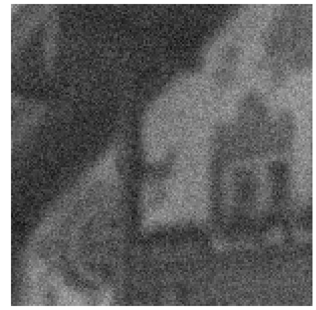
We have also performed the same experiment for other image deblurring algorithms including Wiener algorithm [18] and BM3D deblurring [2] and have obtained the same results: applying grid warping algorithm after deblurring improves the image quality.

V. Conclusion

In this paper we have shown that image sharpening by the grid wapring algorithm is an effective method for postprocessing for linear motion deblurring algorithms.

The warping procedure does not amplify noise or introduce ringing artifact. If an image deconvolution algorithm has a parameter that specifies a balance between low risk of artifacts and better sharpness, it is promising to firstly apply an image deconvolution algorithm with a preference of a low risk of artifacts and then perform edge sharpening by the grid warping algorithm.

(a) Reference image



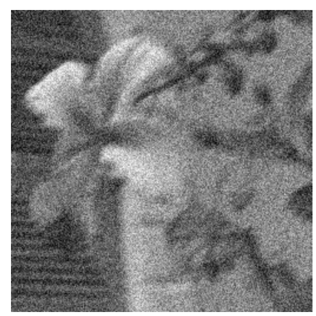
(b) Blurred image. PSNR = 16.83, SSIM = 0.5614

(c) Deconvolution. PSNR = 18.57, SSIM = 0.7606

(d) Deconvolution + warping. PSNR = 18.63, SSIM = 0.7664

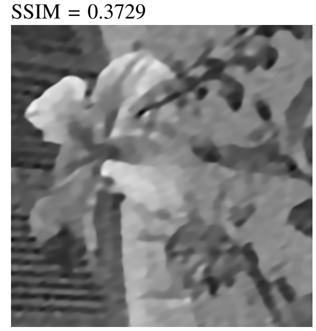
Fig. 6. Application of the deconvolution and warping algorithms to 'house' image fragment, the blur parameters are: $r = 3$, $\ell = 2$, $\sigma_n = 16$.

(a) Reference image



(b) Blurred image. PSNR = 18.05, SSIM = 0.3729

(c) Deconvolution. PSNR = 20.27, SSIM = 0.7520



(d) Deconvolution + warping. PSNR = 20.27, SSIM = 0.7555

Fig. 7. Application of the deconvolution and warping algorithms to 'flower' image fragment, the blur parameters are: $r = 2$, $\ell = 10$, $\sigma_n = 32$.

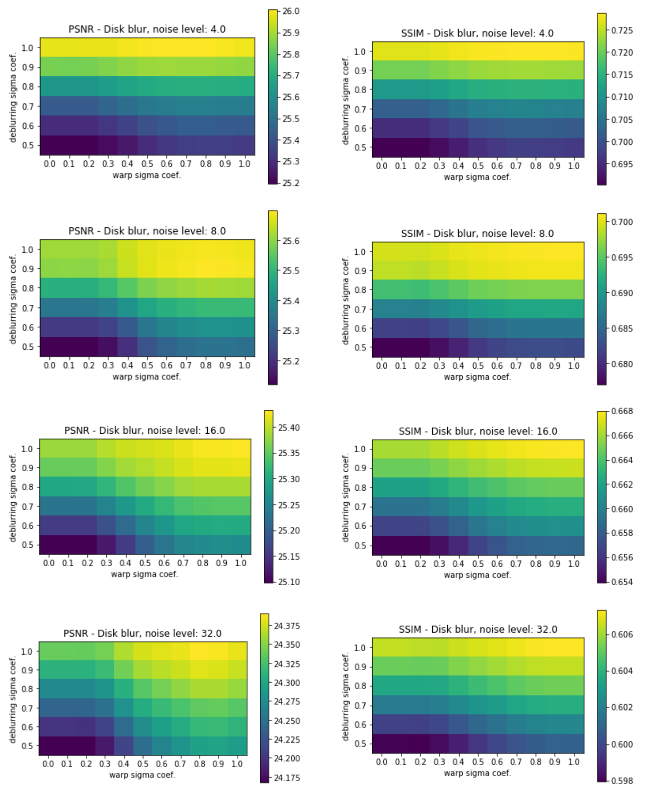


Fig. 8. The dependence of average PSNR and SSIM values on q_B and q_W for difference noise levels.

Acknowledgment

The work was supported by Russian Science Foundation Grant 17-11-01279.

References

[1] J. Oliveira, J. M. Bioucas-Dia, and M. Figueiredo, "Adaptive total variation image deblurring: A majorization-minimization approach," *Signal Process.*, vol. 89, pp. 1683–1693, 2009.

[2] A. Danielyan, V. Katkovnik, and K. Egiazarian, "BM3D frames and variational image deblurring," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1715–1728, 2012.

[3] K. Bredies and T. Valkonen, "Inverse problems with second-order total generalized variation constraints," *Proceedings of SampTA*, vol. 201, 2011.

[4] P. Campisi and K. Egiazarian, *Blind image deconvolution: theory and applications*. CRC press, 2016.

[5] M. Almeida and M. Figueiredo, "Parameter estimation for blind and non-blind deblurring using residual whiteness measures," *IEEE Trans. Image Processing*, vol. 22, pp. 2751–2763, 2013.

[6] S. D. Babacan, R. Molina, and A. K. Katsaggelos, "Variational bayesian blind deconvolution using a total variation prior," *IEEE Trans. Image Process.*, vol. 18, pp. 12–26, 2009.

[7] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar, "Fast and robust multiframe super resolution," *IEEE transactions on image processing*, vol. 13, no. 10, pp. 1327–1344, 2004.

[8] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum, "Image deblurring with blurred/noisy image pairs," in *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3, 2007, p. 1.

[9] J. Sun, W. Cao, Z. Xu, J. Ponce *et al.*, "Learning a convolutional neural network for non-uniform motion blur removal," in *CVPR*, 2015, pp. 769–777.

[10] J. P. Oliveira, M. A. Figueiredo, and J. M. Bioucas-Dias, "Parametric blur estimation for blind restoration of natural images: linear motion and out-of-focus," *IEEE Transactions on Image Processing*, vol. 23, no. 1, pp. 466–477, 2014.

[11] Y. Bahat, N. Efrat, and M. Irani, "Non-uniform blind deblurring by reblurring," in *CVPR*, 2017, pp. 3286–3294.

[12] N. Arad and C. Gotsman, "Enhancement by image-dependent warping," *IEEE Transactions on Image Processing*, vol. 8, pp. 1063–1074, 1999.

[13] J. Prades-Nebot et al., "Image enhancement using warping technique," *IEEE Electronics Letters*, vol. 39, pp. 32–33, 2003.

[14] A. Krylov, A. Nasonova, and A. Nasonov, "Image warping as an image enhancement post-processing tool," *OGRW2014*, pp. 132–135, 2015.

[15] A. Nasonova and A. Krylov, "Deblurred images post-processing by Poisson warping," *IEEE Signal Processing Letters*, vol. 22, no. 4, pp. 417–420, 2015.

[16] A. S. Krylov, A. A. Nasonova, and A. V. Nasonov, "Image enhancement by non-iterative grid warping," *Pattern Recognition and Image Analysis*, vol. 26, no. 1, 2016.

[17] N. Ponomarenko, L. Jin, O. Ieremeiev, V. Lukin, K. Egiazarian, J. Astola, B. Vozel, K. Chehdi, M. Carli, F. Battisti, and C.-C. J. Kuo, "Image database TID2013: Peculiarities, results and perspectives," *Signal Processing: Image Communication*, vol. 30, pp. 57–77, 2015.

[18] J. G. Nagy, K. Palmer, and L. Perrone, "Iterative methods for image deblurring: A matlab object-oriented approach," *Numerical Algorithms*, vol. 36, no. 1, pp. 73–93, 2004.