

# Deblurring noisy radial-blurred images: spatially adaptive filtering approach

Giacomo Boracchi<sup>a,b</sup>, Alessandro Foi<sup>c</sup>, Vladimir Katkovnik<sup>c</sup>, and Karen Egiazarian<sup>c</sup>

<sup>a</sup>Dipartimento di Elettronica ed Informazione,  
Politecnico di Milano, via Ponzio, 34/5, 20133 Milano  
E-mail: giacomo.boracchi@polimi.it

<sup>b</sup>Tampere International Center for Signal Processing,  
Tampere University of Technology, P.O. Box 553, 33101 Tampere, Finland.

<sup>c</sup>Department of Signal Processing,  
Tampere University of Technology, P.O. Box 553, 33101 Tampere, Finland.  
E-mail: firstname.lastname@tut.fi

## ABSTRACT

The deblurring of images corrupted by radial blur is studied. This type of blur appears in images acquired during any camera translation having a substantial component orthogonal to the image plane. The point spread functions (PSF) describing this blur are spatially varying. However, this blurring process does not mix together pixels lying on different radial lines, i.e. lines stemming from a unique point in the image, the so called “blur center”. Thus, in suitable polar coordinates, the blurring process is essentially a 1-D linear operator, described by the multiplication with the blurring matrix.

We consider images corrupted simultaneously by radial blur and noise. The proposed deblurring algorithm is based on two separate forms of regularization of the blur inverse. First, in the polar domain, we invert the blurring matrix using the Tikhonov regularization. We then derive a particular modeling of the noise spectrum after both the regularized inversion and the forward and backward coordinate transformations. Thanks to this model, we successfully use a denoising algorithm in the Cartesian domain. We use a non-linear spatially adaptive filter, the Pointwise Shape-Adaptive DCT, in order to exploit the image structures and attenuate noise and artifacts.

Experimental results demonstrate that the proposed algorithm can effectively restore radial blurred images corrupted by additive white Gaussian noise.

## 1. INTRODUCTION

In this work we consider the restoration of images corrupted by blur produced by a camera that translates in the 3D space with constant velocity during the acquisition. This situation can be formalized as follows. Let  $\mathbf{u}$  be the 3D vector that identifies the camera translation during the exposure time  $T$ . This translation can be decomposed in two components:  $\mathbf{u}_\perp$  and  $\mathbf{u}_0$ , which are orthogonal and parallel to the image plane respectively, as shown in Figure 1(a). Typically, image restoration algorithms assume  $\mathbf{u}_\perp = 0$ , considering the camera translating parallel to the image plane. This assumption leads to the spatially invariant blur degradation model (see <sup>1-3</sup> and references therein).

In this paper we focus on the generic case when  $\mathbf{u}_\perp \neq 0$ . The presence of a significant component  $\mathbf{u}_\perp$  makes the blur spatially variant and the image restoration becomes a much more challenging problem. For simplicity, the captured scene is assumed planar and parallel to the image plane, thus neglecting complications due to the scene depth.

---

This work was partially supported by CIMO, the Finnish Centre for International Mobility (fellowship TM-07-4952) and by the Academy of Finland (project No. 213462, Finnish Centre of Excellence program 2006 - 2011).

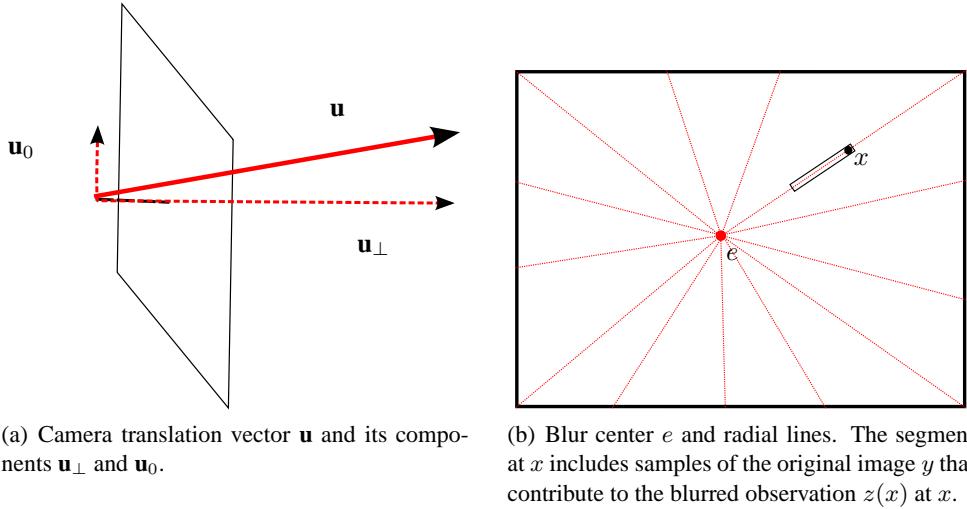


Figure 1. Radial blurred image formation.

Let us consider the blurred image  $z$  as integration of *sub-images*  $y_t$ ,

$$z(x) = \int_0^T y_t(x) dt, \quad x = (x_1, x_2) \in \mathcal{X}, \quad (1)$$

where  $x$  represents a coordinate in the image domain  $\mathcal{X}$ ,  $T$  is the exposure time and  $y_t$  is a sub-image, i.e. a sharp image produced by the light intensity that reaches the sensor at the instant  $t \in [0, T]$ . Each sub-image  $y_t$  is acquired with a different viewpoint along the camera trajectory.

In the trivial case when  $u_{\perp} = 0$ , all sub-images are shifted w.r.t. each other, i.e.  $y_t(x) = y_0(x - \lambda t)$ ,  $\lambda \in \mathbb{R}^2$ . Because the shift  $\lambda t$  is the same for each point  $x$ , the blurred image can be modeled as the convolution of the original image with a kernel, the point spread function (PSF), which has a 1D support and which is typically parametrized by its direction and its extent.

When the camera translation is not parallel to the image plane, i.e.  $u_{\perp} \neq 0$ , the integration support corresponding to each point varies. The relations between any of two sub-images are described by the essential matrix,<sup>4</sup> which acts differently on each image point. Nevertheless, in this case, the blur can be described by the blur center  $e$  and the blur extent parameter  $l$ . Such images are termed *radial blurred images* because the blur smears are directed along radial lines, i.e. lines stemming from the blur center, as shown in Figure 1(b). As we move away from the blur center, which itself is not blurred, the length of the smears grows with rate equal to the extent  $l$ . In particular, the smear at a point  $x$  such that  $|x - e| = 1$  has length  $l$ , where  $|x - e|$  is the distance between pixel  $x$  and the blur center  $e$ .

There are only few publications about the restoration of images corrupted by the radial blur. Webster and Reeves<sup>5</sup> addressed this problem and proposed a fast restoration algorithm based on resampling the blurred image on a lattice where the blur becomes spatially invariant, so that any deconvolution technique (e.g., in Fourier domain) can be used. However, in their work, the image noise was not considered explicitly. In principle, any method for restoring images corrupted by spatially variant blur<sup>6</sup> could be used for radial blur. However, these methods result in heavy computational costs, while the restoration of radial blurred images becomes simpler after performing adequate image transformations. In fact, when the image is transformed in polar coordinates w.r.t. to the blur center  $e$ , the blurring process having extent  $l$  can be modeled by the multiplication against an upper triangular matrix  $B_l$ .

The restoration algorithm proposed in this paper exploits Cartesian-to-polar and polar-to-Cartesian coordinate transformations which enable a computationally affordable blur inversion. Because a naive pseudo-inverse can amplify the noise and artifacts, we exploit two separate forms of regularization in both polar and Cartesian domains. In the polar domain, we invert the matrix  $B_l$  using a Tikhonov regularized inverse. Then, we perform denoising in the Cartesian domain using a non-linear spatially adaptive filter, the Pointwise Shape-Adaptive DCT,<sup>7-9</sup> in order to exploit the image structures and

attenuate both noise and artifacts. For this filtering we derive approximate models of the noise spectrum for the forward and backward coordinate transformations, in order to drive the denoising in Cartesian image domain.

The rest of the paper is organized as follows. Section 2 presents a model for images corrupted by radial blur, Section 3 describes the blur inversion stage in the ideal, noiseless case. The noise is introduced and handled in Section 4, where the noise characteristics after the blur inversion are estimated in order to be applied in the denoising algorithm. Finally, in Section 5 we validate the proposed algorithm by experimental results.

## 2. IMAGE FORMATION

Typically, a spatially variant blurred image  $z$  is modeled by the following integral

$$z(x) = \int_{\mathcal{X}} k(x, s)y(s)ds, \quad x = (x_1, x_2) \in \mathcal{X} \quad (2)$$

where  $y$  is the original image and the PSF  $k(x, \cdot)$  determines how the intensity values of the original image  $y$  contribute in  $z(x)$ .

In the case of the radial blur, the PSF in each image pixel  $x$  is determined by its position w.r.t. the blur center  $e = (e_1, e_2)$  and by the extent parameter  $l$ . The blur center  $e$  is the pixel that corresponds to the image of the vanishing point of camera translation direction, and thus it is the only pixel which is not blurred in the radial blurred image. The blur extent parameter  $l > 0$  determines how the blur extent increases along each radial line, i.e. the blur extent at a pixel  $x$  is  $l|x - e|$ . The parameter  $l$  depends on the length of the camera displacement and on the distance between the camera viewpoint and the planar scene (measured along the camera translation direction).

Figure 3(a) shows an example of a radial blurred image: the blur is visibly characterized by smears along the radial lines. The radial blurring process does not mix together pixels lying on different radial lines and thus the radial blur can be compactly formulated in the polar coordinates.

Let  $\mathfrak{P}$  and  $\mathfrak{C}$  be two continuous domain transforms that perform respectively Cartesian-to-polar and polar-to-Cartesian image mapping w.r.t. the blur center  $e$ :

$$Z(\rho, \theta) = \mathfrak{P}(z)(\rho, \theta) = z(e_1 + \rho \cos \theta, e_2 + \rho \sin \theta), \quad (3)$$

and

$$z(x_1, x_2) = \mathfrak{C}(Z)(x_1, x_2) = Z\left(\sqrt{(x_1 - e_1)^2 + (x_2 - e_2)^2}, \arctan\left(\frac{x_2 - e_2}{x_1 - e_1}\right)\right). \quad (4)$$

Here and in what follows, capital letters indicate images expressed in the polar coordinates, while small case letters indicate images in Cartesian coordinates.

Then, the radial blur can be written as

$$Z(\rho, \theta) = \int_{\mathbb{R}} \frac{1}{\rho l} \chi_{[\rho, \rho+l\rho]}(r) Y(r, \theta) dr, \quad (5)$$

where  $\chi_{[a,b]}$  is the characteristic function of the interval  $[a, b]$ ,

$$\chi_{[a,b]}(r) = \begin{cases} 1 & a \leq r \leq b \\ 0 & \text{else,} \end{cases} \quad (6)$$

In practice we have to deal with discrete data, therefore we use also two discrete domain coordinate transformations,  $\mathcal{P}$  and  $\mathcal{C}$ , which perform the Cartesian-to-polar and the polar-to-Cartesian mapping, respectively. These transformations can be obtained by discretization of the continuous operators of Equations (3) and (4). The transform  $\mathcal{P}$  maps the input  $w \times h$  image into an  $r \times \tau$  output matrix. To obtain such an output with a rectangular domain in polar coordinates, the transform implicitly pads the input image (e.g., by zero-padding). Without loss of generality and for practical reasons, we assume that the blur center  $e$  is at the image center, as this situation can be reproduced by padding and shifting the image adequately.

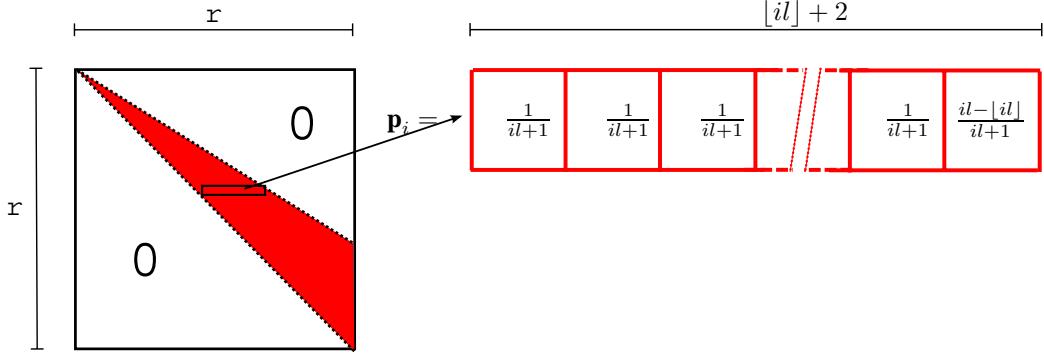


Figure 2. The blurring matrix  $B_l$  is an upper triangular  $r \times r$  matrix. The upper part presents only a narrow band above the diagonal where it is not null. The rows are defined in formula (11) and the non-zero are shaded.

The radial blurred image in the polar coordinates  $Z$  can be expressed as a matrix multiplication

$$Z = B_l Y, \quad (7)$$

where  $Y = \mathcal{P}(y)$  and  $B_l$  is an upper triangular matrix, that depends on the blur extent parameter  $l$ . The noise-free observations are given by

$$z = \mathcal{C}(B_l Y). \quad (8)$$

We emphasize that the above equations are discrete approximations of the continuous domain equations (3) - (5). Note also that both  $\mathcal{P}$  and  $\mathcal{C}$  exploit data interpolation in order to compute image values corresponding to non-integer coordinates, hence it may happen that

$$\mathcal{C}(\mathcal{P}(z)) \neq z \quad \text{and} \quad \mathcal{P}(\mathcal{C}(Z)) \neq Z. \quad (9)$$

## 2.1 The Blurring Matrix

The matrix multiplication between the image in polar coordinates  $Y$  and the blurring matrix  $B_l$  corresponds to a discretization of the operator of Equation (5), which models the radial blurring process of extent  $l$ . Note that in polar coordinates each column of  $Y$  contains  $r$  pixels lying on the same radial line, thus  $B_l$  is a  $r \times r$  matrix

$$B_l = \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \vdots \\ \mathbf{p}_{r-1} \\ \mathbf{p}_r \end{bmatrix}, \quad (10)$$

where the row vectors  $\mathbf{p}_i = \{p_{i,j}\}_{j=1,\dots,r}$ ,  $i = 1, \dots, r$ , are

$$\mathbf{p}_i = \begin{cases} 1/(il+1) & i \leq j \leq \lfloor i + il \rfloor \\ (il - \lfloor il \rfloor)/(il+1) & j = \lfloor i + il + 1 \rfloor, \\ 0 & \text{else,} \end{cases} \quad (11)$$

and  $\lfloor \cdot \rfloor$  denotes the rounding to the nearest smaller or equal integer.

As shown in Figure 2,  $B_l$  is an upper triangular matrix and each row  $\mathbf{p}_i$  corresponds to an horizontal blur PSF, which, provided that the PSF support is totally included within the matrix, has extent  $il + 1$ . These PSFs are exactly the sub-pixel discretization of the function  $\chi$  of integral (5). Note that the last rows of  $B_l$  have  $\ell^1$ -norm smaller than 1, because the supports of the corresponding PSFs lie partially outside of the image domain. In practice, this is equivalent to zero-padding the image and producing the blur with unrestricted PSFs (hence having their norm equal to 1). It results in synthetically blurred images having borders fading to zero, as shown in Figure 3(a).



(a) Radial blurred image: the blur center  $e$  is at the image center and  $l = 0.25$ .

(b) Blur inversion with naive inverse  $B_l^{-1}$ , RMSE = 11.84.

(c) Regularized inverse  $z^{RI}$  via Tikhonov regularization of  $\tilde{B}_l$  (12), RMSE = 11.41.

Figure 3. Blur inversion with naive inverse and Tikhonov regularized inverse.

### 3. BLUR INVERSION

The blur inversion consist of estimating the original image  $y$  from  $z$ , assuming that the parameters  $e$  and  $l$  are known. The observation model presented in Section 2 relies on two image transformations, these transformations are used also for blur inversion. In polar coordinates the blur is a multiplication against the blurring matrix  $B_l$  (7). Therefore, a straightforward solution to blur inversion consists of transforming the observation  $z$  in polar coordinates w.r.t.  $e$ , i.e.  $Z = \mathcal{P}(z)$  and by multiplying it against the inverse of blurring matrix,  $B_l^{-1}$ .

Even if matrix  $B_l$  is definite positive, and thus invertible, the inversion of the blur inevitably amplifies errors due to imprecision in the modeling. Therefore we replace the naive inverse  $B_l^{-1}$  with the regularized Tikhonov inverse matrix  $\tilde{B}_l$ ,

$$\tilde{B}_l = (B_l^T B_l + \alpha^2 I)^{-1} B_l^T, \quad (12)$$

where  $\alpha > 0$  is a regularization parameter. The regularized inverse of the blurred observation is then

$$z^{RI} = \mathcal{C}(\tilde{B}_l \mathcal{P}(z)). \quad (13)$$

Figure 3 shows the blur inversion for the blurred image depicted in Figure 3(a). In Figure 3(b) and Figure 3(c) we can see the inverse by multiplication with the matrix  $B_l^{-1}$  and with the matrix  $\tilde{B}_l$ , respectively. Both images show artifacts along some rows and columns, but these artifacts are stronger in Figure 3(b) than in Figure 3(c). These artifacts are due to the rounding errors in the coordinate transformations  $\mathcal{P}$  and  $\mathcal{C}$ . The root mean square error (RMSE) of the image restored with the naive inversion is 11.84, while in the case of Tikhonov regularization ( $\alpha = 0.005$ ) the RMSE is 11.41. The regularized inverse is able to reduce these artifacts and it is crucial for dealing with image noise.

### 4. NOISE AND NOISE ATTENUATION

For simplicity, we model the observation errors as an additive white Gaussian noise (AWGN) term in the Cartesian domain. The observation equation is as follows

$$z = \mathcal{C}(B_l Y) + \eta, \quad (14)$$

where  $\eta(\cdot) \sim \mathcal{N}(0, \sigma^2)$ .

Figure 4 illustrates the blur inversion on a noisy image. The Gaussian noise has been added to the blurred image according to Equation (14), with a standard deviation  $\sigma = 0.004 \times 255$ , which is a very low noise and almost not perceptible in Figure 4(a). However, as one can clearly see in Figure 4(b), in the regularized inverse  $z^{RI}$  the noise has been significantly amplified and the restoration performance decreased (RMSE= 19.94).



(a) Radial blurred noisy image: the blur center  $e$  corresponds to the image center,  $l = 0.25$ , and  $\sigma = 0.004 \times 255$ . Noise is hardly perceptible in  $z$ .

(b) The regularized inverse  $z^{RI}$  presents heavy noise, RMSE = 19.94.

(c) Final restored image, obtained after Pointwise SA-DCT filtering of the regularized inverse  $z^{RI}$  shown in Figure 4(b): RMSE = 12.81.

Figure 4. Algorithm performance on a radial blurred image  $z$  with noise.

Note that the noise characteristics change significantly after the coordinate transformations and the multiplication against the inverse of the blurring matrix. Even when the noise in  $z^{RI}$  is Gaussian distributed, which happens if  $\mathcal{P}$  and  $\mathcal{C}$  exploit linear interpolation, the noise may not be independent nor identically distributed.

#### 4.1 Image Restoration Algorithm

Our image restoration algorithm is based on a two step approach like many state-of-the-art deconvolution algorithms.<sup>9–12</sup> The first step is the blur inversion. In the cited works, this consists of regularized deconvolution in Fourier domain. The noise in the deconvolved image is Gaussian colored noise and it is characterized by its power spectral density (PSD), which (up to a  $\sigma^2$  scaling factor) is determined by the PSF. The second step is the removal of the colored noise. This is performed by conventional filtering techniques, such as transform-domain shrinkage.<sup>9–11</sup> In the cited works the noise PSD is used for computing threshold for the shrinkage.

In our case, the blur inversion consists of multiplying the image in polar domain against the matrix  $\widetilde{B_l}$ . Because of the coordinate transformations and because the blur is not convolutional, the noise in  $z^{RI}$  cannot simply be described by a PSD. However, we treat the noise in  $z^{RI}$  as colored Gaussian and we use an approximate model for the noise PSD. The denoising is then performed by the Pointwise Shape-Adaptive DCT (SA-DCT) filtering algorithm.<sup>7–9</sup> This algorithm had been used earlier for Gaussian colored noise removal in deblurring<sup>9</sup> and inverse halftoning applications.<sup>13</sup>

#### 4.2 Noise PSD Modeling

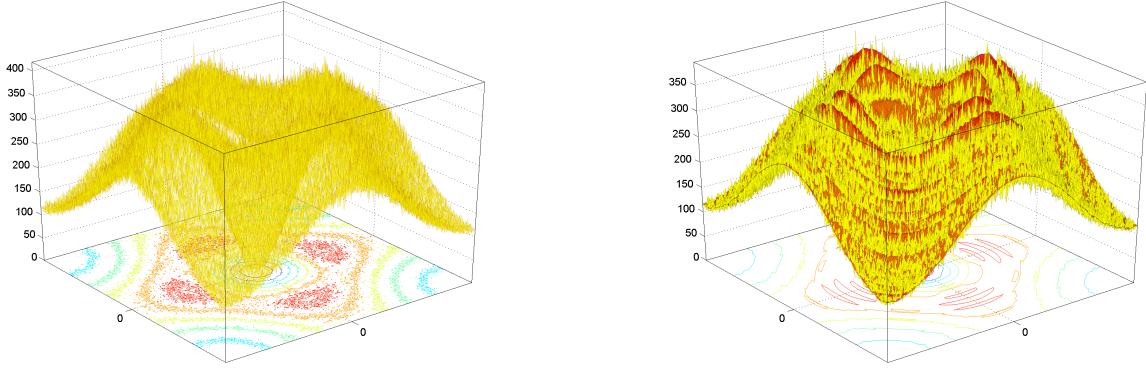
Calculating the noise statistics in  $z^{RI}$  is a demanding task as  $\mathcal{P}$  and  $\mathcal{C}$  transform differently the image pixels, according to their location on the image plane. Moreover, any analytical result would depend on the interpolation methods used in  $\mathcal{P}$  and  $\mathcal{C}$ . In what follows, we consider the transforms sequence  $\mathcal{C}(\widetilde{B_l}\mathcal{P}(\cdot))$  (i.e. Cartesian-to-polar, blur inversion, and polar-to-Cartesian) as an input-output system and we study the noise statistics by a Monte Carlo approach. That is, we generate  $n$  independent realizations of standard Gaussian white noise  $\eta_i$ ,  $\eta_i(\cdot) \sim \mathcal{N}(0, 1)$ ,  $i = 1, \dots, n$ , and, assuming  $e$  at the image center, we process each of them with the input-output system  $\mathcal{C}(\widetilde{B_l}\mathcal{P}(\cdot))$ . More precisely, let  $\eta_i$  be the system input,

$$\eta'_{i,l} = \mathcal{C}(\widetilde{B_l}\mathcal{P}(\eta_i)), \quad i = 1, \dots, n, \quad (15)$$

be the system output. In what follows, we restrict ourselves to operators  $\mathcal{P}$  and  $\mathcal{C}$  which are linear.

The noise PSD  $N_l$  is computed as the sample variance of the Fourier transforms of the  $n$  realizations of  $\eta'_{i,l}$ ,  $i = 1, \dots, n$  along each frequency:

$$N_l(\omega) = \text{var}\{\mathcal{F}(\eta'_{i,l})(\omega)\}, \quad \omega \in \Omega, \quad (16)$$



(a) The noise PSD  $N_{0.25}$ . Its values have been computed using Equations (15) and (16).  
(b) The function  $S_{0.25}$  (in red), approximating  $N_{0.25}$  of Figure 5(a) (in yellow).

Figure 5. The noise PSD  $N_{0.25}$  and the approximating function  $S_{0.25}$ .

where  $\mathcal{F}$  denotes the 2D Fourier transform and  $\omega$  represents the frequencies in the Fourier domain  $\Omega$ . We remark that  $N_l$  is computed from  $n$  realizations of noise with unitary variance. Whenever the actual noise has variance  $\sigma^2 \neq 1$ , the noise PSD needs to be scaled accordingly and becomes  $\sigma^2 N_l$ . Note also that the computed PSD depends on the blur extent  $l$ .

By taking into account the PSD for the denoising, we implicitly treat the noise in  $z^{RI}$  as Gaussian colored noise. This may not necessarily hold, however this approximate noise description facilitates the application of the image denoising algorithm. In practice, this pragmatic approach yields satisfactory results from experimental evidence.

Figure 5(a) presents the noise PSD  $N_{0.25}$ , obtained by inverting  $n = 3000$  noise realizations with radial blur of extent  $l = 0.25$ . One can see that despite obvious symmetries within the PSD, the middle horizontal/vertical cross-sections are considerably different from the two diagonal ones. These cross-sections, which we denote  $v_l$  (horizontal/vertical) and  $d_l$  (diagonals) are shown in Figure 7.

Since the noise PSD varies depending on  $l$ , we repeat the Monte Carlo procedure considering several different values of the blur extent parameter. Furthermore, for practical purposes, we replace the sampled  $N_l$  with an approximated function  $S_l$  generated by convex interpolation of the horizontal/vertical and diagonal cross-sections  $v_l$  and  $d_l$  of  $N_l$ . In our simulations the blur extents are  $l = 1/120, \dots, 110/120$ . Figure 8 illustrates the cross-sections  $v_l$  and  $d_l$  when varying the blur extent  $l$ .

Finally  $S_l$ , the function approximating  $N_l$ , is defined as

$$S_l(\omega) = \cos^2(2\beta)S_l^v(\omega) + \sin^2(2\beta)S_l^d(\omega), \quad \omega \in \Omega, \quad (17)$$

where  $\beta = \arctan(\frac{\omega_2}{\omega_1})$  is the angular component of the frequency  $\omega$  and  $S_l^v$  and  $S_l^d$  are surfaces of revolution generated rotating around the origin  $v_l$  and  $d_l$ , respectively. Figures 6(a) and 6(b) show the surfaces  $S_{0.25}^v$  and  $S_{0.25}^d$ , respectively, while Figure 6(c) shows  $S_{0.25}$  computed according to Equation (17). The quality of the approximation of  $S_{0.25}$  to  $N_{0.25}$  is illustrated in Figure 5(b).

Figure 4(c) illustrates the image restoration performance of proposed approach when  $S_{0.25}$  is used as the noise PSD for the Pointwise SA-DCT denoising. The RMSE of the restored image is 12.81.

## 5. EXPERIMENTS

We present simulation results obtained for a set of four common grayscale test images of size  $256 \times 256$ . As in Equation (14), the blurred noisy observation are generated from the original image  $y$  as

$$z = \mathcal{C}(B_l \mathcal{P}(y)) + \eta. \quad (18)$$

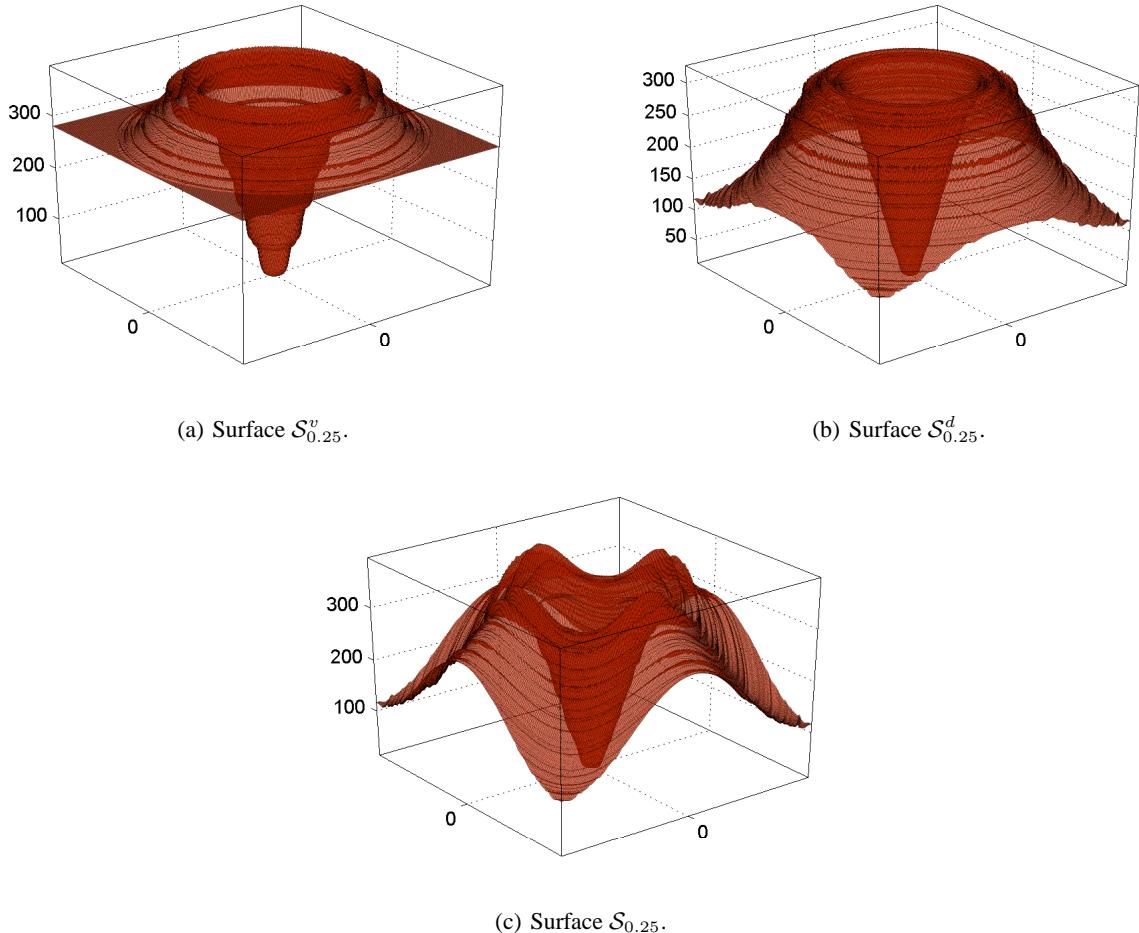
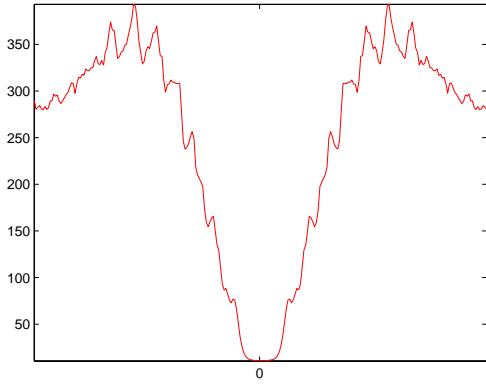


Figure 6. Top: examples of surfaces of revolution  $S_l^v$  and  $S_l^d$  generated from the cross-sections  $v_l$  and  $d_l$ . Bottom:  $S_l$ , approximation of the noise PSD  $N_l$ , obtained as the convex combination of  $S_l^v$  and  $S_l^d$  defined by Equation (17).

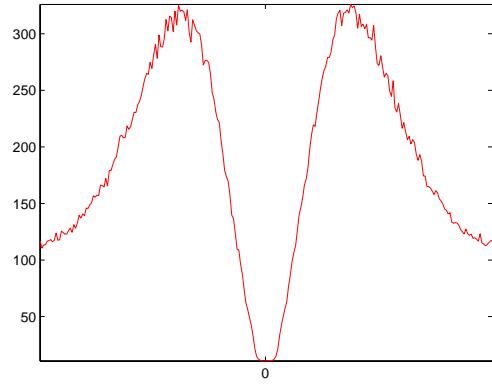
We use discrete transforms  $\mathcal{P}$  and  $\mathcal{C}$  based on bilinear interpolation. Let us remark that both these transforms introduce errors in the observation  $z$ , seriously impairing the restoration quality even in the noise free case. The size of the polar domain is determined as in the work by Ribaric *et al.*<sup>14</sup> on images blurred because of camera rotation. In particular, we use  $r = \max_{x \in \mathcal{X}}(|x - e|)$  and  $\tau = \lceil 2\pi / \arcsin(\sqrt{2}/r) \rceil$ , where  $\lceil \cdot \rceil$  is the rounding to the nearest larger or equal integer. We limited our tests to blur having the blur center  $e$  at the image center: whenever  $e$  is in a different location on the image plane, in order to apply the restoration algorithm, the image has to be accordingly padded and shifted.

According to a common practice when testing deblurring algorithms, we add noise with a small variance to images where the blur extent is large and noise with a higher variance to images where the blur extent is small. Thus, we mimic the situation where images are acquired with different exposure times during the same camera motion. When the camera undergoes a fixed translation, images acquired with a long exposure are typically heavily blurred, while the noise affecting these images is small. On the contrary, images acquired with a short exposure, show weaker blur and stronger noise. Table 1 shows the pairs blur extent/noise standard deviation used for generating the blurred noisy observations. For all these cases, the Tikhonov regularization parameter,  $\alpha$  (12), is fixed to  $\alpha = 0.005$ .

Table 2 shows the RMSE of the restored images w.r.t. the original image. The performance of the noise attenuation are illustrated in Figure 9, while Figure 10 shows some of the images, before and after restoration. As one may expect, the

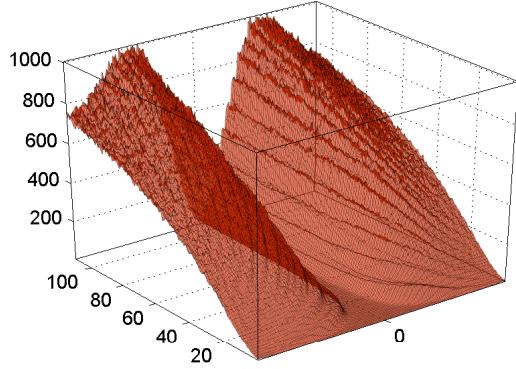


(a) The values  $v_{0.25}$  corresponding to the PSD values on the vertical axis of  $\Omega$ .

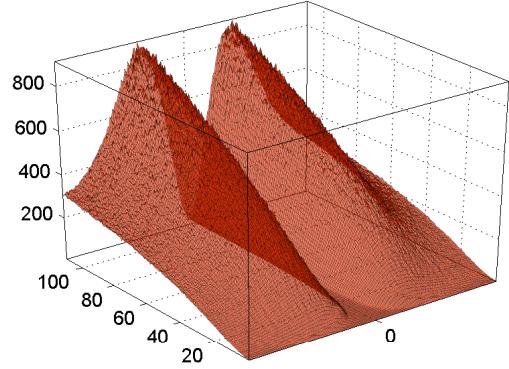


(b) The values  $d_{0.25}$  corresponding to the PSD values on the diagonal of  $\Omega$ .

Figure 7. Examples of  $v_l$  and  $d_l$  generating the surfaces  $\mathcal{S}_l^v$  and  $\mathcal{S}_l^d$ .



(a) The noise PSD values at frequencies on the vertical axis of  $\Omega$  when varying the parameter  $l = 1/120, \dots, 110/120$ .



(b) The noise PSD values at frequencies on the diagonals of  $\Omega$  when varying the parameter  $l = 1/120, \dots, 110/120$ .

Figure 8. PSD values used for computing  $\mathcal{S}_l$ , with  $l = 1/120, \dots, 110/120$ .

algorithm performance decreases with the increase either of the blur extent  $l$  or of the noise standard deviation. The latter appears to have a more substantial impact on the quality of the restored image. In particular, the RMSE corresponding to Exp.6, which is the noise-free experiment, is always significantly lower than the RMSE in Exp.5, where the observations are generated with the same blur but with  $\sigma = 0.002 \times 255$ . Finally, let us observe that the restored images of Figure 10 show some artifacts along the radial lines; these are not due to noise but rather to interpolation errors introduced by the coordinate transformations  $\mathcal{P}$  and  $\mathcal{C}$ : indeed such artifacts appear also in the noise-free experiments.

Table 3 gives the execution times of each separate stage of the algorithm, applied on a  $256 \times 256$  image (Cameraman). The times correspond to our Matlab implementation running on a computer with AMD 64 1.81-GHz processor. As one can see from the Table 3, the impact of coordinate transformations on the overall execution time is of negligible, as nearly all the time is actually taken by the Pointwise SA-DCT denoising step.

Experiment	$60l$	$\sigma/255$
Exp.1	5	0.006
Exp.2	10	0.004
<b>Exp.3</b>	<b>15</b>	<b>0.004</b>
Exp.4	20	0.002
Exp.5	25	0.002
Exp.6	25	0

Table 1. Experimental settings: blur extent and noise standard deviation values used for testing the proposed algorithm. The third row (in bold) shows the parameters used in the examples of Figures 3 and 4.

	Exp.1	Exp.2	Exp.3	Exp.4	Exp.5	Exp.6
Cameraman	11.27	11.98	12.82	12.24	12.52	12.33
House	6.93	6.93	7.33	6.50	6.63	6.94
Aerial	11.13	11.43	12.34	10.69	11.09	9.43
Peppers	8.79	8.81	9.35	8.46	8.63	8.42

Table 2. Root mean squared error (RMSE) of each restored image in the experimental settings of Table 1. Examples of the restoration quality are shown in Figure 10.

## 6. CONCLUSIONS

In this work we presented a novel restoration algorithm for noisy radial blurred images. The restoration algorithm includes two main steps: the blur inversion in the polar domain, and the noise removal in the Cartesian domain. The denoising part consists of an adaptation of a spatially adaptive transform-based denoising method, namely the Pointwise SA-DCT filter.<sup>7,9</sup>

Experimental results with synthetically generated observations show that the denoising step improves significantly the restoration performance. At the same time, it emerges the need of an accurate model of noise statistics after both blur inversion and coordinate transformations. In the future we will investigate algorithms for estimating  $e$  and  $l$  from a given blurred image, in order to combine the estimation and the restoration procedures together into a blind deblurring algorithm.

## REFERENCES

1. Q. L. Feng Xue and J. Froment, “An a contrario approach for parameters estimation of a motion-blurred image,” in *Lecture Notes in Computer Science, Energy Minimization Methods in Computer Vision and Pattern Recognition*, S. B. Heidelberg, ed., pp. 267–279, 2007.
2. I. M. Rekleitis, “Steerable filters and cepstral analysis for optical flow calculation from a single blurred image,” in *Vision Interface*, pp. 159–166, (Toronto, Canada), May 1996.
3. Y. Yitzhaky, R. Milberg, S. Yohae, and N. S. Kopeika, “Comparison of direct blind deconvolution methods for motion-blurred images,” *Applied Optics* **38**, pp. 4325–4332, July 1999.
4. R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, second ed., 2004.
5. C. B. Webster and S. J. Reeves, “Radial deblurring with ffts,” in *Proc. IEEE International Conference on Image Processing, ICIP*, **1**, pp. I–101–I–104, (San Antonio, USA), September 2007.
6. J. G. Nagy and D. P. O’Leary, “Restoring images degraded by spatially variant blur,” *SIAM Journal on Scientific Computing* **19**, pp. 1063–1082, July 1998.
7. A. Foi, V. Katkovnik, and K. Egiazarian, “Pointwise shape-adaptive dct for high-quality denoising and deblocking of grayscale and color images,” *IEEE Transactions on Image Processing* **16**(5), pp. 1395–1411, May 2007.
8. A. Foi and K. Dabov, “Pointwise shape-adaptive DCT demobox.” <http://www.cs.tut.fi/~foi/SA-DCT>.
9. A. Foi, K. Dabov, V. Katkovnik, and K. Egiazarian, “Shape-adaptive DCT for denoising and image reconstruction,” in *Proc. SPIE Electronic Imaging: Algorithms and Systems V*, **6064A-18**, (San Jose, CA, USA), January 2006.
10. J. A. Guerrero-Colon, L. Mancera, and J. Portilla, “Image restoration using space-variant gaussian scale mixtures in overcomplete pyramids,” *IEEE Transactions on Image Processing* **17**, pp. 27–41, January 2008.
11. R. Neelamani, H. Choi, and R. Baraniuk, “Forward: Fourier-wavelet regularized deconvolution for ill-conditioned systems,” *IEEE Transactions on Image Processing* **52**, pp. 418–433, February 2004.

Algorithm part	time (sec)
Cartesian-to-polar	0.28
Blur inversion	0.07
Polar-to-Cartesian	0.12
SA-DCT filtering	4.91

Table 3. Execution times for a  $256 \times 256$  test image on a AMD 64 1.81-GHz.

12. V. Katkovnik, K. Egiazarian, and J. Astola, “A spatially adaptive nonparametric regression image deblurring,” *IEEE Transaction on Image Processing* **14**, pp. 1469–1478, October 2005.
13. K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Inverse halftoning by pointwise shape-adaptive DCT regularized deconvolution,” in *Proc. International TICSP Workshop Spectral Methods Multirate Signal Process.*, (Florence, Italy), September 2006.
14. S. Ribaric, M. Milani, and Z. Kalafatic, “Restoration of images blurred by circular motion,” in *Proc. of the First International Workshop on Image and Signal Processing and Analysis, IWISPA*, pp. 53–60, (Pula, Croatia), 14-15 June 2000.

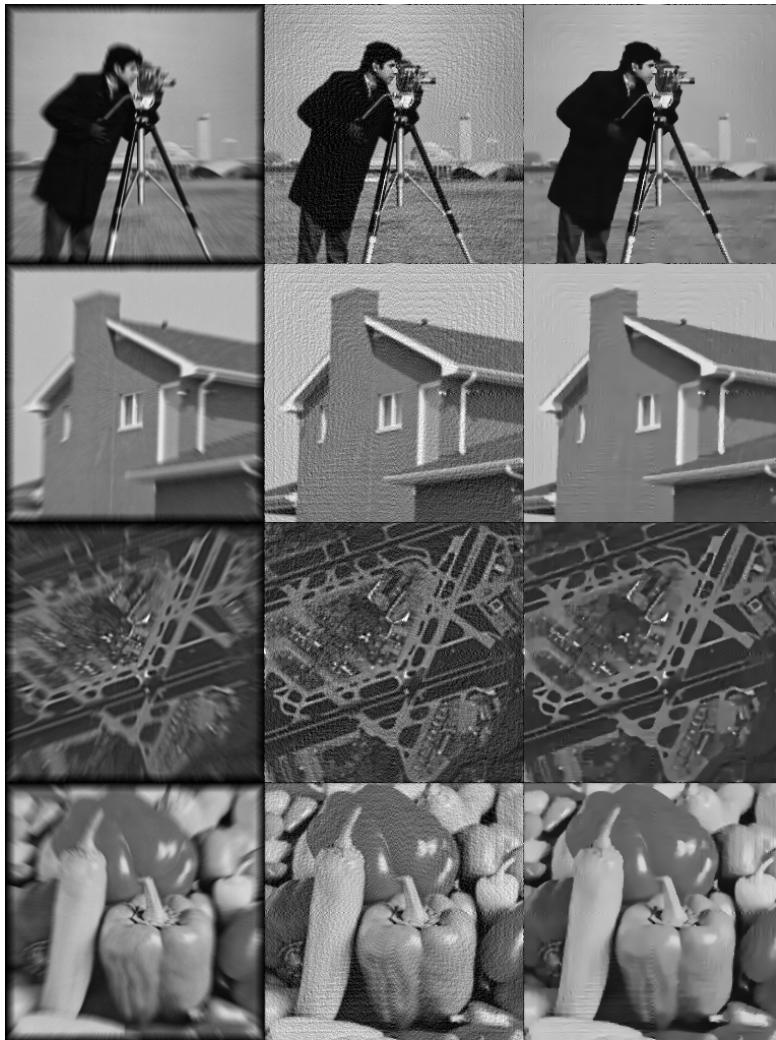


Figure 9. Experiment 1: noise attenuation. The first column shows the observations, the second column the corresponding regularized inverses  $z^{RI}$ , and the third column shows the restored images. Restoration performance are listed in Table 2.



Figure 10. Blurred and noisy images (first and third column) and restored images (second forth column) obtained with the proposed method. See Table 1 for parameters used in the experiments and Table 2 for restoration performance in terms of RSME. Row 1: “Cameraman” experiments 2 and 6. Row 2: “House” experiments 2 and 3. Row 3: “House” experiments 4 and 5. Row 4: “Peppers” experiments 3 and 4. Row 5: “Peppers” experiments 5 and 6. Row 6: “Aerial” experiments 2 and 3. Row 7: “Aerial” experiments 4 and 5.