Journal of Mathematical Imaging and Vision (2018) 60:692–706
https://doi.org/10.1007/s10851-017-0784-7

Algorithm and Constraints for Exact Non-blind Deconvolution

Yechiel Lamash[1]

Received: 24 September 2016 / Accepted: 12 December 2017 / Published online: 10 January 2018

Abstract
Image filtering is generally an irreversible process. Image restoration methods, such as inverse filtering or other deconvolution techniques, cannot precisely recover the original image and often introduce some level of artifacts. In the current paper, we formulate the filtering as a blending of several transformed replications of the original image. We assume zero noise after the filtering. Using this convention, we propose a method that precisely restores the original image from its filtered version. The method is applicable for a family of filters including average box filters and approximated Gaussian filters.

Keywords Non-blind deconvolution · Filter theory · Inverse filtering · Image restoration

1 Introduction

As the Fourier transforms of common low-pass filters are not nonzero everywhere, they are not invertible. Discrete Fourier transform (DFT)-based inverse filtering [15] and deconvolution methods [2,11] are commonly used for estimating the latent image from blurred and noisy images. However, these methods do not produce precise restoration of the original image and often introduce artifacts. Nevertheless, for zero noise after the blurring and under specific constraints on the filter weights and the image's number of bits, we show that the original image can still be restored from its filtered version. Bilateral transition between the original image and its smoothed version is a plausible feature that can be used in many image processing applications. For example, in image editing [1], users might want to 'undo' their smoothing filter or replace it with an edge-preserving denoising method [3,20]. In general, users may have different preferences in the level of sharpness and noise. The ability to return to the original image is generally better than sharpening the smoothed image. In the current study, we show how a family of filters can be fully reversed to generate the original image to machine precision. We also highlight the method's constraints and limitations.

1.1 Related Work

There are several common methods for estimating the original image from its smoothed version when the filter kernel is given. Inverse filtering [15] is the simplest and fastest method that uses the DFT of the kernel and filtered image to estimate the original image. A threshold is generally used to replace zero or smaller DFT values to avoid infinities or extremely high values in the invertible DFT kernel elements. The main disadvantage of this method is that it does not perform well in the presence of noise. The Wiener filter [11] offers an optimal trade-off, in terms of mean square error, between inverse filtering and noise smoothing. However, when the blurring filter is singular, the Wiener filter actually amplifies the noise. Wavelet-based denoising scheme [8] combines Fourier-domain inverse filtering and wavelet-domain image denoising in turn. Although the approach improves the performance, in the case when the blurring function is not invertible, the algorithm is not applicable. Furthermore, since the two steps are separated, there is no control over the overall performance of the restoration. Neelamani et al. [17] proposed a wavelet-based deconvolution technique for ill-conditioned systems that employs both Fourier-domain Wiener-like and wavelet-domain regularization. Several recent methods use priors based on natural image statistics to regularize the ill-posed problem of non-blind deconvolution. A common approach that gives good results in a reasonable time is to assume Laplacian distribution marginals that allow a number of fast $L1$ and related TV-norm methods to be deployed [19,22]. Wang et al. [22]

✉ Yechiel Lamash
shilikster@gmail.com

[1] Computational Radiology Laboratory, Boston Children's Hospital, Harvard Medical School, Boston, MA, USA

used minimization technique known as half-quadratic splitting, originally proposed by Geman and colleagues [9,10] and showed how it could be used with a total-variation (TV) seminorm to deconvolve images. However, real-world images have shown to have heavier tails than a Laplacian. Levin et al. [14] and Joshi et al. [12] have applied hyper-Laplacian image priors to non-blind deconvolution problems using iteratively re-weighted least squares (IRLS) [21] to solve for the deblurred image. Krishnan and Fergus [13] proposed an efficient scheme using a similar approach. Their algorithm uses an alternating minimization scheme where the non-convex part of the problem is solved in one phase, followed by a quadratic phase, which can be efficiently solved in the frequency- domain using FFTs.

In the current study, we propose a method for exact non-blind deconvolution. We assume zero noise after the filtering. The method has a different approach in restoring the original image. Instead of inverting the kernel, we formulate the filtered image as a blending of translated replications of the original image. Using this formulation, our method iteratively eliminates these replications. To the best of our knowledge, this is the first paper to show a method and constraints for exact non-blind deconvolution. The method also obtains superior results for a small amount of additive Gaussian noise, when filtering with average kernels, compared to classical and state-of-the-art non-blind deconvolution methods.

The current paper is organized as follows: Section 2 starts with describing the method of restoring the original image from average filtered image. Since the 2D reversible filter is separable, the restoration is performed in each direction separately, using one-dimensional filters, one direction after the other. For simplicity, we start with one-dimensional kernel formulation of the method. Next, the method is extended to higher- dimensional kernels and to approximated Gaussian filters. Thereafter, we provide the proof for reversible filters of kernels with two nonzero weights. Section 3 illustrates the method and compares it to inverse filtering and to other deconvolution methods. Section 4 highlights the method's limitations. A list of optional applications is suggested in Sect. 5, while Sect. 6 provides the summary and conclusions.

2 Method

We start with formulating a moving average filter in one direction as a blending of several replications of the original image with a constant translation between successive replications. The method's idea is to iteratively update the image by subtracting its own transformed replications while generating new distal ones. The original image is fully restored after the last new-generated replication crosses the image limits. Figure 1 provides a one-dimensional illustration of the method. The current section provides the mathematical formulation of the problem, the proposed solution and a proof of the method's bit exact image restoration.

2.1 Reversing a Moving Average Filter

Given the input filtered image with the known kernel size as an input, and the above formulation, we will now define the problem of obtaining the original image.

Problem statement
Given an image $\hat{I}(x)$ that consists of an average of multiple equal images with a fixed relative translation

$$\hat{I}(x) = \frac{1}{N}\sum_{i=0}^{N-1} I(x-i) \tag{1}$$

where $N > 1$ is the filter size.

We would like to obtain the original image $I(x)$.

For simplicity, in this section, we omit the y index since the processing is one dimensional. In the following sections, x denotes a two-dimensional vector of coordinates.

Proposition *Denote the image size by Ω. Applying the following operation $k \geq \lceil log_2(\Omega/N)\rceil$ times*

$$\hat{I}_{k+1}(x) = \begin{cases} \hat{I}_k(x) - \hat{I}_k(x-1), & k = 1 \\ \hat{I}_k(x) + \hat{I}_k(x - 2^{k-2}N), & k > 1 \end{cases} \tag{2}$$

will result in

$$\hat{I}_{k+1}(x) = \frac{1}{N} I(x) - \frac{1}{N} I(x - 2^{k-1}N) \tag{3}$$

Since for $k \geq \lceil log_2(\Omega/N)\rceil + 1$ the second term $I(x-\Omega)$ is outside the image limits for all pixels $0 < x < \Omega$, we get

$$I(x) = N\hat{I}_{k+1}(x). \tag{4}$$

Proof We prove the proposition using induction. First, we show that for $k = 1, 2, 3$ the proposition holds.

For $k = 1$, we get

$$\begin{aligned}\hat{I}_2(x) &= \hat{I}_1(x) - \hat{I}_1(x-1) \\ &= \frac{1}{N}\sum_{i=0}^{N-1} I(x-i) - \frac{1}{N}\sum_{i=1}^{N} I(x-i) \\ &= \frac{1}{N} I(x) - \frac{1}{N} I(x-N)\end{aligned} \tag{5}$$

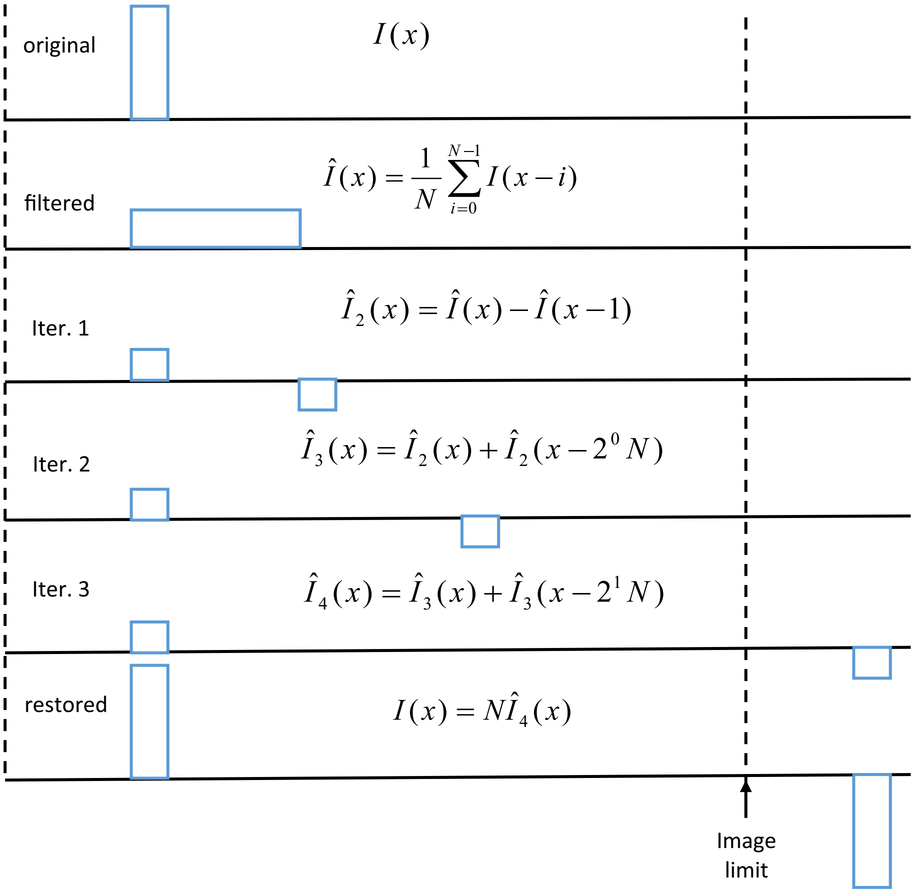

original
$I(x)$
filtered
$\hat{I}(x)=\frac{1}{N}\sum_{i=0}^{N-1} I(x-i)$
Iter. 1
$\hat{I}_2(x)=\hat{I}(x)-\hat{I}(x-1)$
Iter. 2
$\hat{I}_3(x)=\hat{I}_2(x)+\hat{I}_2(x-2^0 N)$
Iter. 3
$\hat{I}_4(x)=\hat{I}_3(x)+\hat{I}_3(x-2^1 N)$
restored
$I(x)=N\hat{I}_4(x)$
Image limit

Fig. 1 One-dimensional illustration of the reversible filtering restoration process. N is the filter size. The first iteration is one-dimensional derivative of the signal. The other iterations remove repetitions when generating a new distal one. When the repetition crosses the image limit, the image is fully restored

Going one step forward to $k = 2$, we get

$$\begin{aligned}\hat{I}_3(x) &= \hat{I}_2(x) + \hat{I}_2(x - 2^0 N)\\ &= \frac{1}{N}I(x) - \frac{1}{N}I(x-N)\\ &\quad + \frac{1}{N}I(x-N) - \frac{1}{N}I(x-2N)\\ &= \frac{1}{N}I(x) - \frac{1}{N}I(x-2N)\end{aligned} \tag{6}$$

For $k = 3$, we get

$$\begin{aligned}\hat{I}_4(x) &= \hat{I}_3(x) + \hat{I}_3(x - 2^1 N)\\ &= \frac{1}{N}I(x) - \frac{1}{N}I(x-2N)\\ &\quad + \frac{1}{N}I(x-2N) - \frac{1}{N}I(x-4N)\\ &= \frac{1}{N}I(x) - \frac{1}{N}I(x-4N)\end{aligned} \tag{7}$$

Assuming that for some $k = n$

$$\hat{I}_{n+1}(x) = \frac{1}{N}I(x) - \frac{1}{N}I(x - 2^{n-1}N) \tag{8}$$

we will show that for $k = n + 1$

$$\hat{I}_{n+2}(x) = \frac{1}{N}I(x) - \frac{1}{N}I(x - 2^{n}N) \tag{9}$$

By applying Eq. (2), we get

$$\hat{I}_{n+2}(x) = \hat{I}_{n+1}(x) + \hat{I}_{n+1}(x - 2^{n-1}N) \tag{10}$$

Setting the induction assumption to the above for each of the two elements, we receive

$$\begin{aligned}\hat{I}_{n+2}(x) &= \frac{1}{N}I(x) - \frac{1}{N}I(x - 2^{n-1}N)\\ &\quad + \frac{1}{N}I(x - 2^{n-1}N)\\ &\quad - \frac{1}{N}I(x - 2^{n-1}N - 2^{n-1}N)\\ &= \frac{1}{N}I(x) - \frac{1}{N}I(x - 2^{n}N)\end{aligned} \tag{11}$$

□

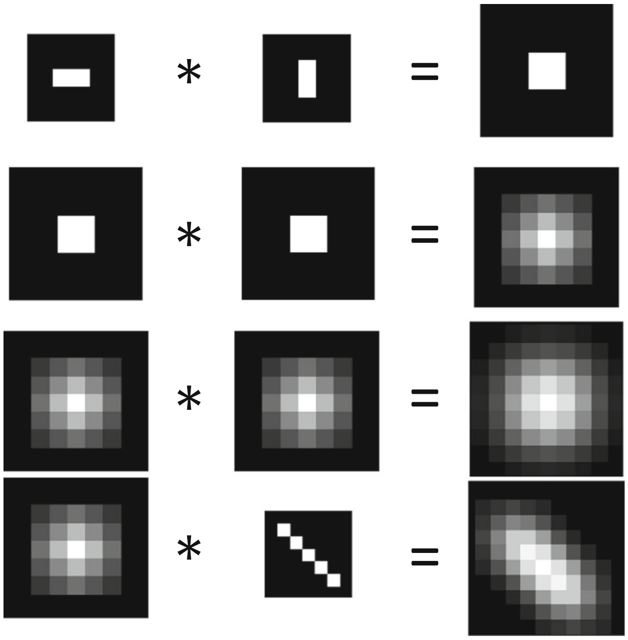


Fig. 2 Convolution of one-dimensional filters to generate two-dimensional box filter and approximated Gaussian filters—obtained using convolution

2.2 Extension to 2D Approximated Gaussian Filters

Until now we have described the exact restoration (reversible filtering) on one-directional average kernel. In the current section, we generalize the reversible filtering to kernels with uneven weights and higher dimension. Since the filtering is formulated using blending of transformed images that can be reversed, we can compose the transformations (or filtrations) and reverse them. Therefore, the key idea is to build the final filter kernel using convolution of one-dimensional average filters. Then, restore the original image by applying the restoration of these one-dimensional average filters (described above in Sect. 2.1), iteratively, one filter after the other. The order of the filters in the convolution or restoration does not matter as the operation is linear. On the other hand, the length and direction of the one-dimensional average filters composing the final filter need to be saved.

We denote these one-dimensional filters by $N_x^{(i)}$ and $N_y^{(j)}$, for the x and y directions, respectively, where i and j are the indices of filters in the x and y directions, respectively.

One implication of the central limit theorem [18] is that convolution of several independent filters approximates a Gaussian filter. This way, the filtering and reverse filtering of approximated Gaussian filters can be performed. Figure 2 demonstrates the filter generation process. Figure 3 demonstrates the iterative process of reverse filtering for two-dimensional average filter.

Figure 4 shows the method's results for two-dimensional average kernel and for two-dimensional approximated Gaussian kernels.

The reverse filtering can be formulated in several ways. Here, we prefer to formulate the translation using the convention of transformation matrix M defined using homogeneous coordinates. This enables simple index writing of x and y as two-dimensional vectors and eases the generalization to linear transformations in Sect. 2.3. For example for translations ty, tx in the vertical and horizontal directions, respectively:

$$M_y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix}, \quad M_x = \begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{12}$$

Therefore, a composition of two vertical translations tx_1, tx_2, will generate a translation of $tx_1 + tx_2$. It is worth mentioning that using homogeneous coordinates, composition is implemented simply using matrix multiplication.

For average reverse filtering, we use translation of one pixel in each direction, i.e., $tx = ty = 1$.

For example, convolving the image with one-dimensional average filter $N_x = 3$ is translated to $\frac{1}{3}(I + I(Mx) + I(M^2x))$

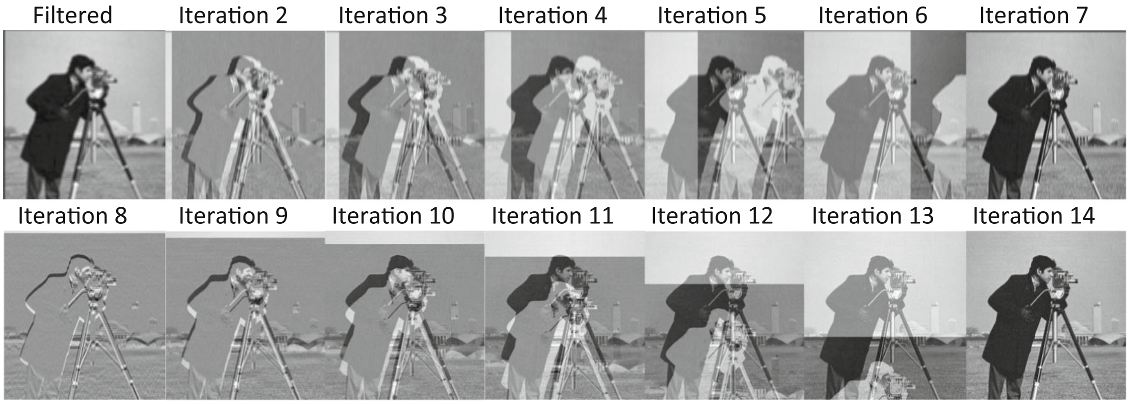


Fig. 3 Illustration of the reversible filtering restoration process. The input image (upper left) was filtered with a [5 × 5] average kernel. The resulted image (lower right) is identical to the original image up to machine precision

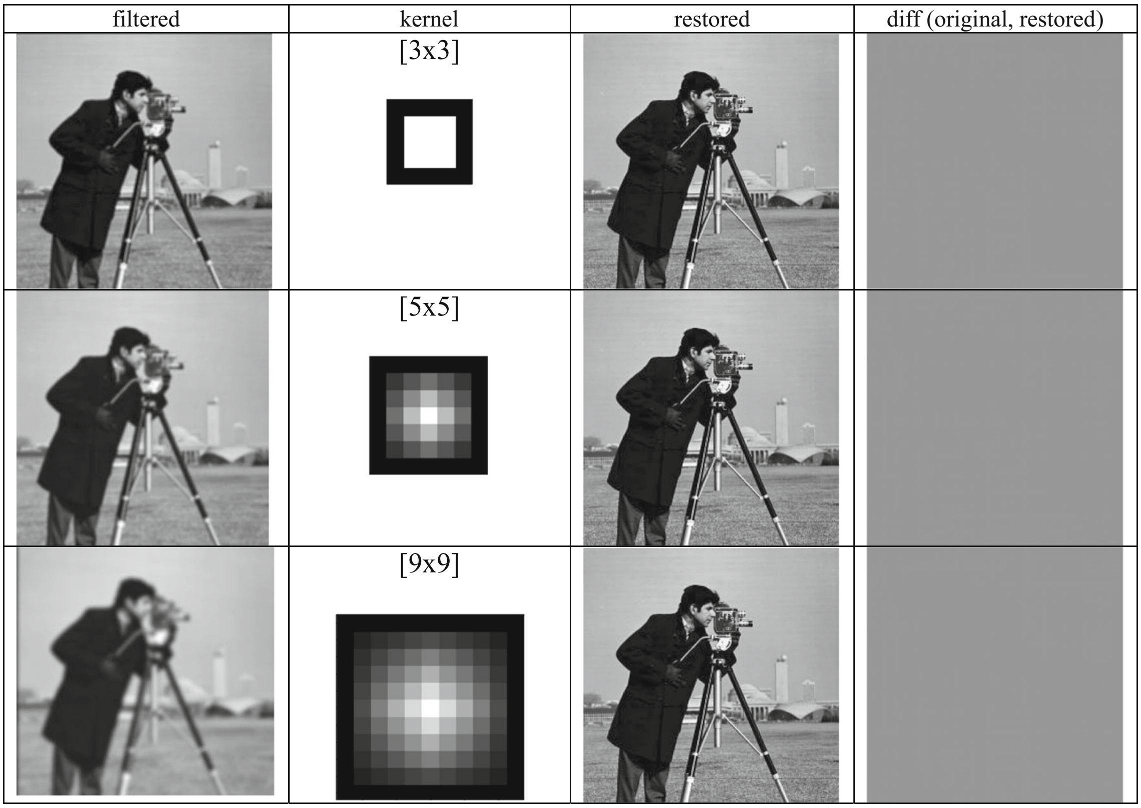


Fig. 4 Illustration of the reversible filtering restoration for two-dimensional average kernel (upper row) and for approximated Gaussian kernels (mid- and lower rows)

where, $M = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ Below is a description of the algorithm stages. As we loop on the x and y directions, the matrix M refers to either M_x or M_y depending on the direction of the current loop. Figure 5 illustrates the method's flow. Below is the pseudo-code of reversing approximated Gaussian filters (denoted by Algorithm A).

```
Algorithm A:
Input: $\hat{I}$, $N_x^{(i)}$, $N_y^{(j)}$
For each one-directional filter of size $N \in \left\{N_x^{(i)}, N_y^{(j)}\right\}$:
generate the transformation matrix $M$
  Loop: $k = 1, \ldots, \lceil \log_2(\Omega/N) \rceil + 1$:
    1) Get the warped image $\hat{I}_k(Mx) = \mathrm{Warp}(\hat{I}_k, M)$
    2) Update $\hat{I}$ using Eq. (2)
    3) Update the transformation matrix:
       if $k = 1 : M \leftarrow M \cdot M^k$ else $M \leftarrow M \cdot M$
```

2.3 Reversible Filters with Two Nonzero Weights

Another type of filters that can be reversed are filters with two nonzero weights. The current section provides the proof for their bit exact reversibility.

We formulate the problem using two replications with relative translation:

Problem statement
Given an image $\hat{I}(x)$ that consists of integration of two equal images with relative translation u

$$\hat{I}(x) = \alpha I(x) + \beta I(x-u) \tag{13}$$

We would like to obtain the original image $I(x)$. We assume that α is not smaller than β. If this is not the case, we switch between the notations of α and β with an opposite sign to u.

Proposition *Denote the image size by Ω.*

Applying the following operation $k \geq \lceil log_2 |\Omega/u| \rceil$ times

$$\hat{I}_{k+1}(x) = \begin{cases} \hat{I}_k(x) - \left(\frac{\beta}{\alpha}\right)^{2^{k-1}} \hat{I}_k(x - 2^{k-1}u), & k = 1 \\ \hat{I}_k(x) + \left(\frac{\beta}{\alpha}\right)^{2^{k-1}} \hat{I}_k(x - 2^{k-1}u), & k > 1 \end{cases} \tag{14}$$

will result in

$$\hat{I}_{k+1}(x) = \alpha I(x) - \frac{\beta^{2^k}}{\alpha^{2^k - 1}} I(x - 2^k u) \tag{15}$$

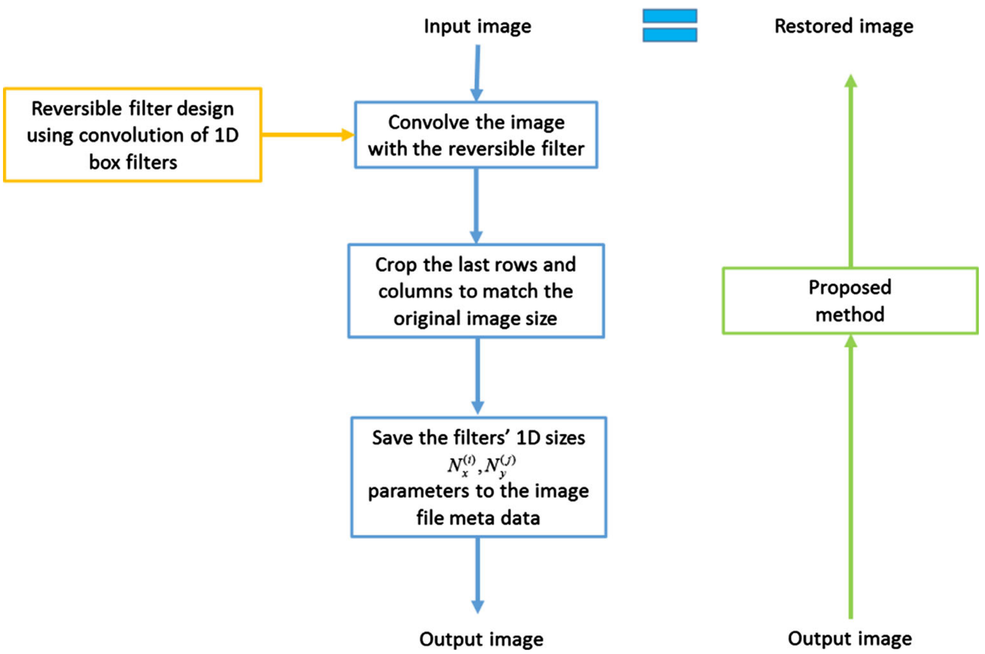

Input image
Restored image
Reversible filter design using convolution of 1D box filters
Convolve the image with the reversible filter
Crop the last rows and columns to match the original image size
Save the filters' 1D sizes $N_x^{(i)}, N_y^{(j)}$ parameters to the image file meta data
Proposed method
Output image
Output image

Fig. 5 Illustration of the method's systematic flow. The downward path (blue) illustrates the filtering process. The reversible filter parameters $N_x^{(i)}$, $N_y^{(j)}$ (i and j denote the number of filters in the x and y directions, respectively) are stored to the meta-data of the image's file. The upward path (green) illustrates the restoration process (Algorithm A, 2.2) (Color figure online)

Since for $k \geq \lceil log_2|\Omega/u|\rceil$ the second term $I(x-2^k u)$ is outside the image limits for all pixels $0 < x < \Omega$ we get

$$I(x) = \hat{I}_{k+1}(x)/\alpha. \tag{16}$$

Proof We prove the proposition by induction. First we show that for $k = 1, 2$ the proposition holds. Setting $k = 1$ to Eq. (14) yields

$$\hat{I}_2(x) = \hat{I}_1(x) - \left(\frac{\beta}{\alpha}\right)^1 \hat{I}_1(x-u) \tag{17}$$

$$\begin{aligned}\hat{I}_2(x) &= \alpha I(x) + \beta I(x-u)\\ &\quad - \left(\frac{\beta}{\alpha}\right)^1 [\alpha I(x-u) + \beta I(x-2u)]\\ &= \alpha I(x) - \frac{\beta^2}{\alpha} I(x-2u)\end{aligned} \tag{18}$$

Going one step forward with $k = 2$, we get

$$\hat{I}_2(x) = \alpha I(x) - \frac{\beta^2}{\alpha} I(x-2u) \tag{19}$$

$$\hat{I}_2(x-2u) = \alpha I(x-2u) - \frac{\beta^2}{\alpha} I(x-4u) \tag{20}$$

$$\begin{aligned}\hat{I}_3(x) &= \hat{I}_2(x) + \left(\frac{\beta}{\alpha}\right)^2 \hat{I}_2(x-2u)\\ &= \alpha I(x) - \frac{\beta^2}{\alpha} I(x-2u)\\ &\quad + \left(\frac{\beta}{\alpha}\right)^2 \left[\alpha I(x-2u) - \frac{\beta^2}{\alpha} I(x-4u)\right]\\ &= \alpha I(x) - \frac{\beta^4}{\alpha^3} I(x-4u)\end{aligned} \tag{21}$$

Assuming that for some $k = n$

$$\hat{I}_{n+1}(x) = \alpha I(x) - \frac{\beta^{2^n}}{\alpha^{2^n-1}} I(x-2^n u) \tag{22}$$

we will show that

$$\hat{I}_{n+2}(x) = \alpha I(x) - \frac{\beta^{2^{n+1}}}{\alpha^{2^{n+1}-1}} I(x-2^{n+1} u) \tag{23}$$

From (14), we get that

$$\hat{I}_{n+2}(x) = \hat{I}_{n+1}(x) + \left(\frac{\beta}{\alpha}\right)^{2^n} \hat{I}_{n+1}(x-2^n u) \tag{24}$$

Setting the assumption both for $\hat{I}_{n+1}(x)$ and for $\hat{I}_{n+1}(x-2^n u)$ and put in (14)

$$\begin{aligned}\hat{I}_{n+2}(x) &= \alpha I(x) - \frac{\beta^{2^n}}{\alpha^{2^n-1}} I(x-2^n u) + \left(\frac{\beta}{\alpha}\right)^{2^n}\\ &\quad \times \left[\alpha I(x-2^n u) - \frac{\beta^{2^n}}{\alpha^{2^n-1}} I(x-2^n u - 2^n u)\right]\end{aligned} \tag{25}$$

After some simple algebra, we get

$$\hat{I}_{n+2}(x) = \alpha I(x) - \frac{\beta^{2^{n+1}}}{\alpha^{2^{n+1}-1}} I(x - 2^{n+1}u) \tag{26}$$

The reader may notice that once $I(x)$ was obtained the translated image $I(x-u)$ can be restored also by applying

$$I(x-u) = \frac{\hat{I}(x) - \alpha I(x)}{\beta} \tag{27}$$

Below is the pseudo-code of reversing filters with two nonzero weights (denoted by Algorithm B). □

```
Algorithm B:
Input: $\hat{I}, M, \alpha, \beta$
Loop: $k = 1, \ldots, \lceil \log_2(\Omega/u) \rceil + 1$ :
  1)Get the warped image $\hat{I}_k(Mx) = Warp(\hat{I}_k(x), M)$
  2) Update $\hat{I}$ using Eq. (14)
  3)Update the transformation matrix: $M \leftarrow M \cdot M$
```

2.4 Restoring an Image from Its Linear Transformed Blended Replications

This section and the subsequent one have nothing to do with filtering, and the restored image is not bit exact with the original one. Yet, the generalization and constraints of restoring the image from its linear transformed replications sum are still an interesting problem. Denote transformation matrix using homogeneous coordinates by M.

For example, affine transformation matrix should look like this:

$$M = \begin{bmatrix} a_1 & a_2 & tx \\ a_3 & a_4 & ty \\ 0 & 0 & 1 \end{bmatrix} \tag{28}$$

Using homogeneous coordinates, the composition of two linear warps is obtained by matrix multiplication. The image $\hat{I}(x)$ consists of a weighted sum of two replications with a relative linear transformation

$$\hat{I}(x) = \alpha I(x) + \beta I(Mx) \tag{29}$$

Since we can isolate the translation component u from the transformation M, applying the following operation $k \geq \lceil log_2 |\Omega/u| \rceil$ times

$$\hat{I}_{k+1}(x) = \begin{cases} \hat{I}_k(x) - \left(\frac{\beta}{\alpha}\right)^{2^{k-1}} \hat{I}_k(Mx), & k = 1 \\ \hat{I}_k(x) + \left(\frac{\beta}{\alpha}\right)^{2^{k-1}} \hat{I}_k(M_k x), & k > 1 \end{cases} \tag{30}$$

where $M_k = M_{k-1} \cdot M_{k-1}$ with $M_1 = M$ will result in

$$\hat{I}_{k+1}(x) = \alpha I(x) - \frac{\beta^{2^k}}{\alpha^{2^k-1}} I\left(M^{2^k} x\right) \tag{31}$$

The translation component of the transformation M^{2^k} is $2^k u$. Therefore, the second term is outside the image limits.

The method can also be extended to linear transformations using composition of linear warps.

Given an image $\hat{I}(x)$ consisting of average of multiple equal images with a fixed relative transformation

$$\hat{I}(x) = \frac{1}{N} \sum_{i=0}^{N-1} I(x - iMx) \tag{32}$$

Since we can isolate the translation component u from the transformation M, applying the below operation $k \geq \lceil \log_2 |\Omega/u| \rceil + 1$ times

$$\hat{I}_{k+1}(x) = \begin{cases} \hat{I}_k(x) - \hat{I}_k(Mx), & k = 1 \\ \hat{I}_k(x) + \hat{I}_k(M_k x), & k > 1 \end{cases} \tag{33}$$

where $M_k = M_{k-1} \cdot M_{k-1}$ with $M_1 = M$ will result in

$$\hat{I}_{k+1}(x) = \frac{1}{N} I(x) - \frac{1}{N} I\left(M^{2^{k-1}} x\right) \tag{34}$$

The translation component of the transformation M^{2^k} is $2^k u$. Therefore, the second term is outside the image limits.

The reason that the restoration is not bit exact is related to the interpolation of the images that cannot be precise in the image edges. A special care of the edges may reduce the generated artifacts in the restored image. Figures 6 and 7 illustrate the restoration from its affine transformed blended replications.

3 Numerical Experiments

To test the performance of the proposed method with respect to alternative restoration algorithms, we compared the results of the current scheme to the inverse filtering [15], Lucy–Richardson [2], Wiener [11], Wang et al. [22] and Krishnan–Fergus [13] deconvolution methods. The results are demonstrated in Figs. 8 and 9.

The algorithms' parameters were: default MATLAB parameters with 10 iterations for the Lucy–Richardson deconvolution and noise-to-signal ratio (NSR) of 0.01 for the Wiener deconvolution. A threshold of 0.0001 was used for the inverse filtering. For the Wang et al. [22] algorithm:

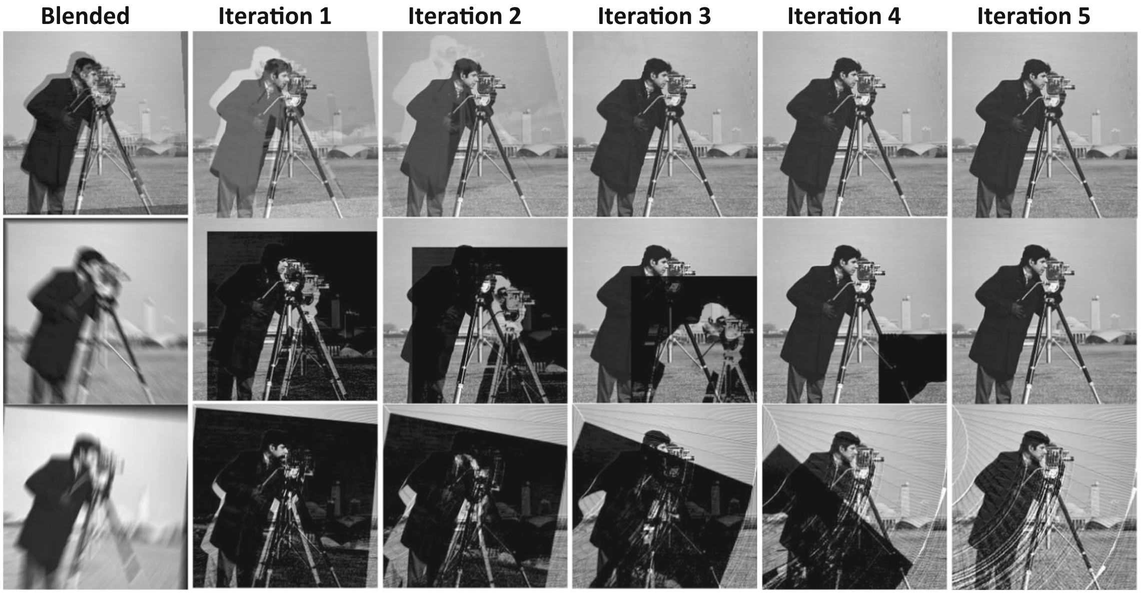


Fig. 6 The method's iterative restoration process from the blended images (left) to the restored images (right). The middle row describe exact restoration

$l2$ was chosen with default values according to [6,7]. For the Krishnan–Fergus algorithm [13]: λ of $2e-3$ and α of $1/2$ were used.

As will be further detailed in Sect. 4, the proposed method requires the preservation of the initial rows and columns after the convolution. Therefore, the last rows and columns are cropped to match the original image size. For the inverse filtering, the blurring was performed in the frequency domain as it yields better restoration results. For the rest of the deconvolution methods, the original image size was preserved by taking the central part of the convolution after filtering with zero-value boundary handling.

The root-mean-square error (RMSE) of the restored image with respect to the original image for the noiseless scenario was computed as a function of the kernel size. The results are illustrated in Fig. 10 and summarized in Table 1.

To demonstrate the advantages and limitations of the proposed method more comprehensively with respect to the other deconvolution methods, the effect of various levels of additive noise after the filtering was tested. The cameraman image with [9 × 9] kernel size was used. The results are shown in Fig. 11. The comparison is also demonstrated in Fig. 12 on a typical image filtered using a [3 × 3] average kernel followed by an additive Gaussian noise with a standard deviation $\sigma = 0.01$. Figure 12 demonstrates the pattern of restoration error and shows that for very small level of noise ($\sigma = 0.01$ corresponds to noise level of 0.004%) the method provides competitive results.

4 Method Limitations

The proposed method requires the filtered image to be stored with sufficient number of bits such that values are not truncated after the filtering. This could be possible either with floating point or with fixed point by multiplying the image with the filter divisor beforehand. Since low light and low dynamic range images are generally noisy, the final image generally undergoes some histogram adjustment. By using a linear histogram adjustment, the reversible filter can be coupled with the linear factor such that the final values are not truncated. The second limitation is that for reversing the filtering, the first rows and columns cannot be cropped from the image file after the filtering. Instead, more of the last rows and columns are cropped to match the original image size. This issue can be bypassed by displaying the filtered image without the first kernel rows and columns. Since the first rows and columns are essential for the restoration, it is important to note that when using border handling other than zero value, the filtered padded lines must also be stored for the restoration of the original image. The third limitation is that for exact restoration the filter kernel must be known beforehand, therefore the method is less feasible for blind deconvolution applications [4].

5 Applications

The current section suggests several optional applications using the proposed method.

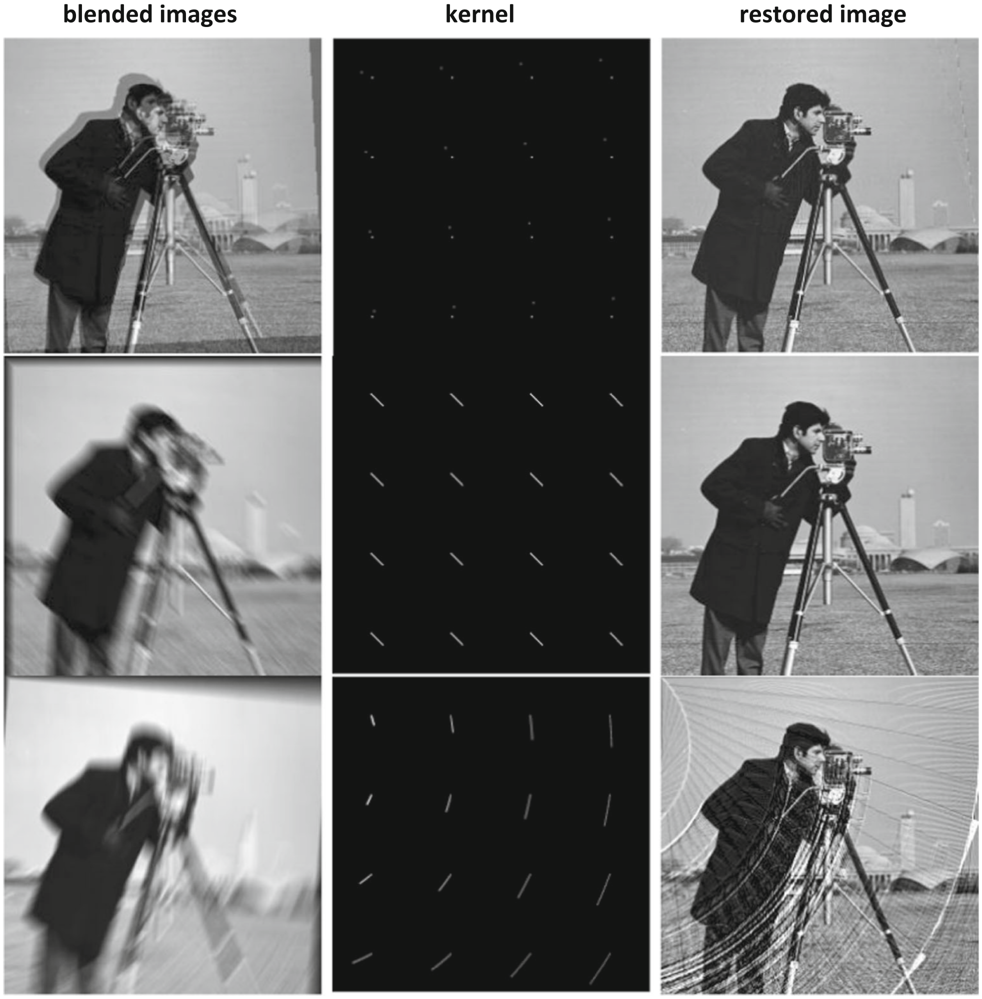


Fig. 7 Illustrative results: the blended images (left), the kernel (mid), and the restored images (right) for the following three blended image simulations: weighted sum of two replications with relative affine transform (top); bit exact restoration for average of multiple translated replications or filtering (middle); and average of multiple ($N = 20$) affine transformed replications (bottom). The artifacts are related to the interpolation at the image edges. The middle column demonstrates the kernel that is spatially variant in affine transformed replications

Table 1 Comparison of the RMSE of the restored image with respect to the original image as a function of the kernel size

	$N = 3$	$N = 5$	$N = 7$	$N = 9$	$N = 11$
Inverse filtering	15.2	22.6	27.8	32.0	35.6
Lucy	31.7	45.3	55.6	63.6	70.0
Wiener	24.2	32.9	37.8	42.6	46.0
Wang et al.	47.5	66.6	95.1	97.6	127.6
Krishnan–Fergus	23.5	37.8	56.5	74.6	87.2
Proposed method	6E–12	1.5E–11	1.8E–11	2.9E–11	3.8E–11

Average kernels were used. Noise was not added after the blurring

5.1 Image Denoising

As state-of-the-art denoising algorithms [3,20] are expensive and of limited use in many real-time applications, sometimes smoothing filters are the only available option. Using the current method, the smoothing filter can be replaced later on with an edge-preserving denoiser if needed or if such denoiser is available. As high frequencies cannot be restored after a regular smoothing filter, the edge-preserving denoiser is expected to perform much better on the original or restored image by preserving the original resolution and edges.

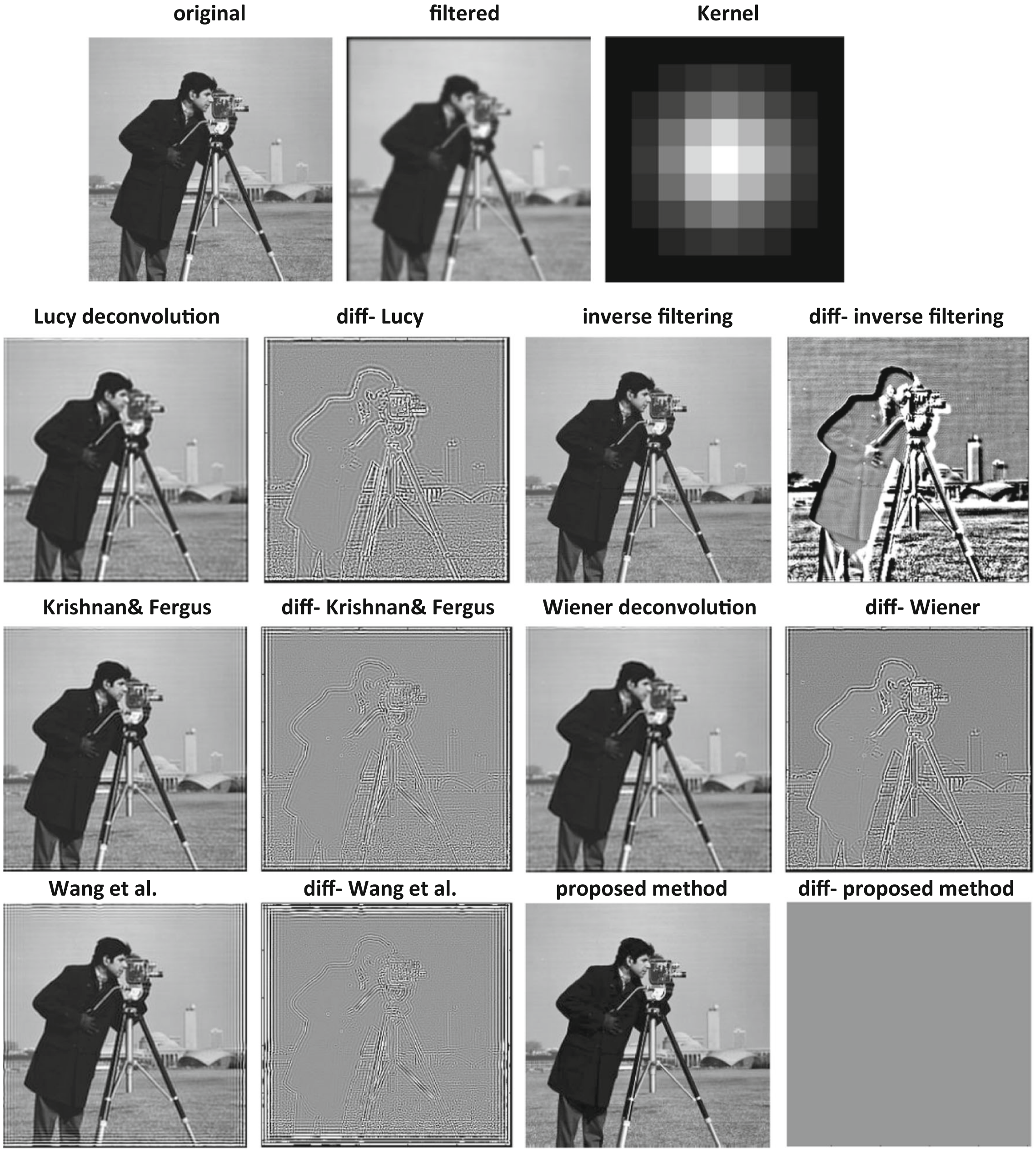


Fig. 8 Comparison of the proposed method to inverse filtering and other deconvolution methods. The first row shows the original image (left), filtered image using the proposed scheme (middle) and the kernel. The rest of the rows show pairs of restored and diff images using the various methods. The diff image color limits are [− 20, 20]

In low-illumination condition, the dark image is noisy and does not use the entire number of bits. Therefore, a reversible filter, to reduce the image noise, can be selected such that if multiplying the maximal value with the filter's denominator, the filtered image values are not truncated. Using the proposed method, the original image can be restored and a different sophisticated denoising method can be used to improve the final image quality. Replacing/canceling the denoising method can also be performed on image formats with extended number of bits (e.g., wide dynamic range image formats) captured in any illumination condition.

5.2 Image Editing of Motion Blur in 2D Computer Graphics

In 2D computer graphics, motion blur is an artistic filter to simulate the visual motion effect. Many graphical software products such as Adobe Photoshop [5] or GIMP [23] offer

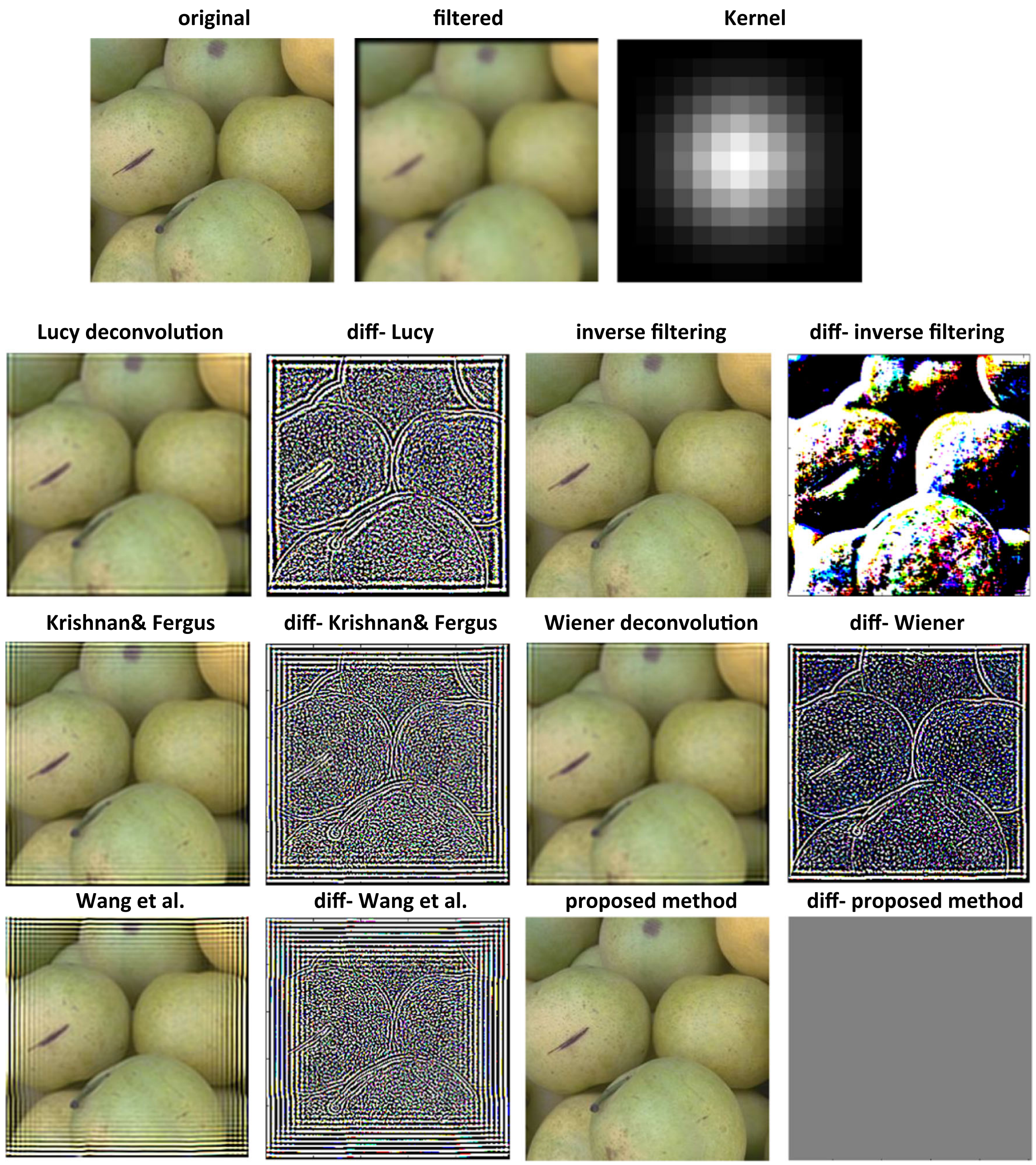


Fig. 9 Comparison of the reverse filtering to inverse filtering and to deconvolution methods. The same restoration process was performed on each color channel. The first row shows the original image (left), filtered image using the proposed scheme (middle) and the kernel. The rest of the rows show pairs of restored and diff images using the various methods. The diff image color limits are [− 20, 20]. RMSE: Inverse filtering (20.2), Lucy (15.6), Wiener (15.9), Krishnan–Fergus (23.6), Wang (72.3), Proposed (0.0)

simple motion blur filters. Using the proposed method, the pre-filtered image can be restored to cancel or modify the blurring direction or strength.

5.3 Control Web Commercial Visual Content

Several web commercial strategies are used to give a customer the ability to estimate the product before making the decision of purchasing it. Watermarking enables visualization of the image while protecting copyright/ownership. Google Books [16] enables the customer to see parts of the book. The proposed method can be used to enable control over the visual content with respect to the purchase status without the need to download the content again. For example, the sharpened high-quality visual version will become available only after purchasing it or after renewing the mem-

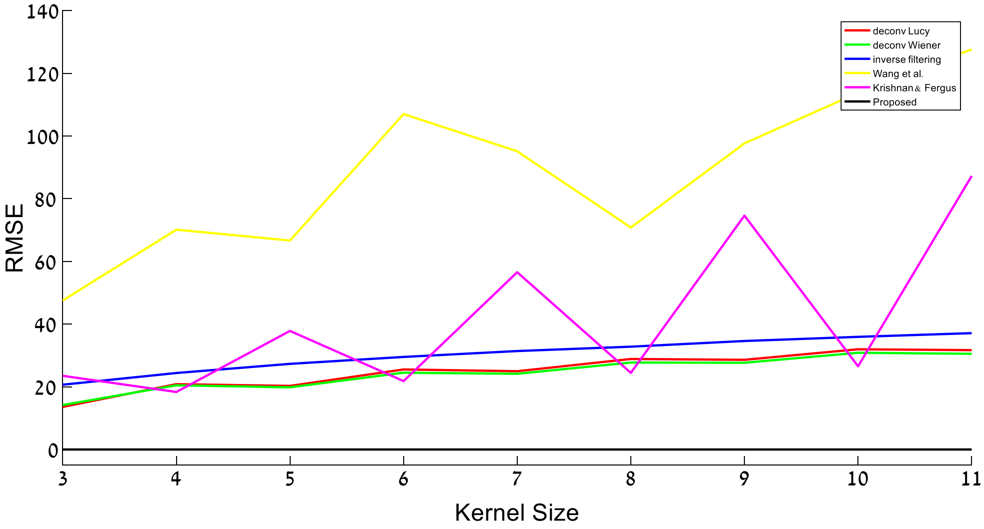


Fig. 10 Comparison RMSE of the restored image with respect to the original image as a function of the kernel size. The comparison was on the cameraman image. Average kernels were used. Noise was not added after the blurring

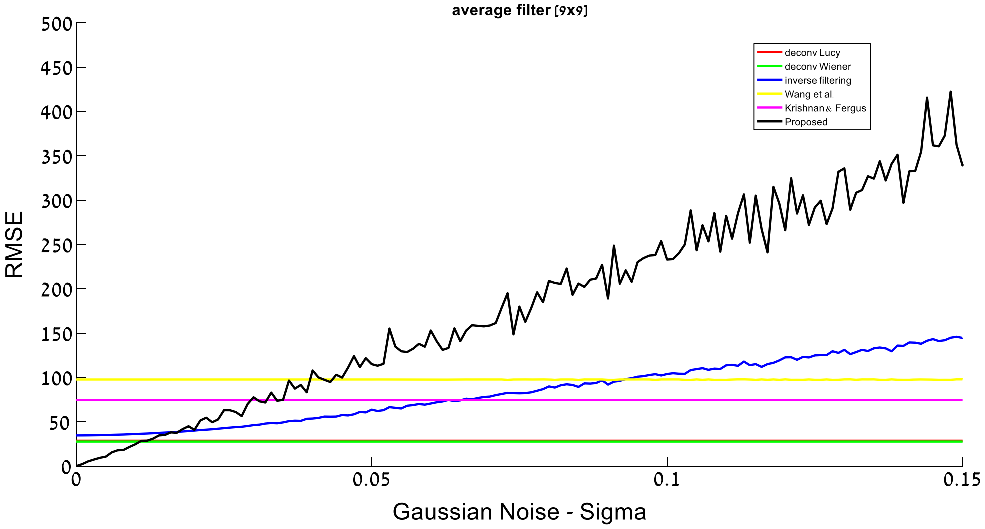


Fig. 11 Comparison RMSE of the restored image with respect to the original image as a function of the Gaussian noise standard deviation—σ. The comparison was on the cameraman image with a [9 × 9] average filter. The proposed method gets the lowest restoration error for $\sigma < 0.01$

bership. This can be implemented by withholding the filters' parameters, which can be unique for each page or image until getting the payment.

5.4 Restoring the Bayer/Color Image from the Luma Channel

In sensors' processing chain, computer vision applications are often computed on the luma (Y channel of the YCbCr color space) rather than on the RGB color image to save hardware resources. Therefore, the input Bayer image is used to generate both the output RGB image and the luma image. This requires memory for storing the Bayer/RGB image during the computation of the computer vision application. The extraction of the luma image can be obtained by filtering the Bayer pattern image with the filter $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}/16$. Since this filter can be calculated from two horizontal and two vertical convolutions of the one-dimensional filter [1, 1]/2, the original Bayer image can be restored from the luma image using the proposed method. In order to avoid truncation of

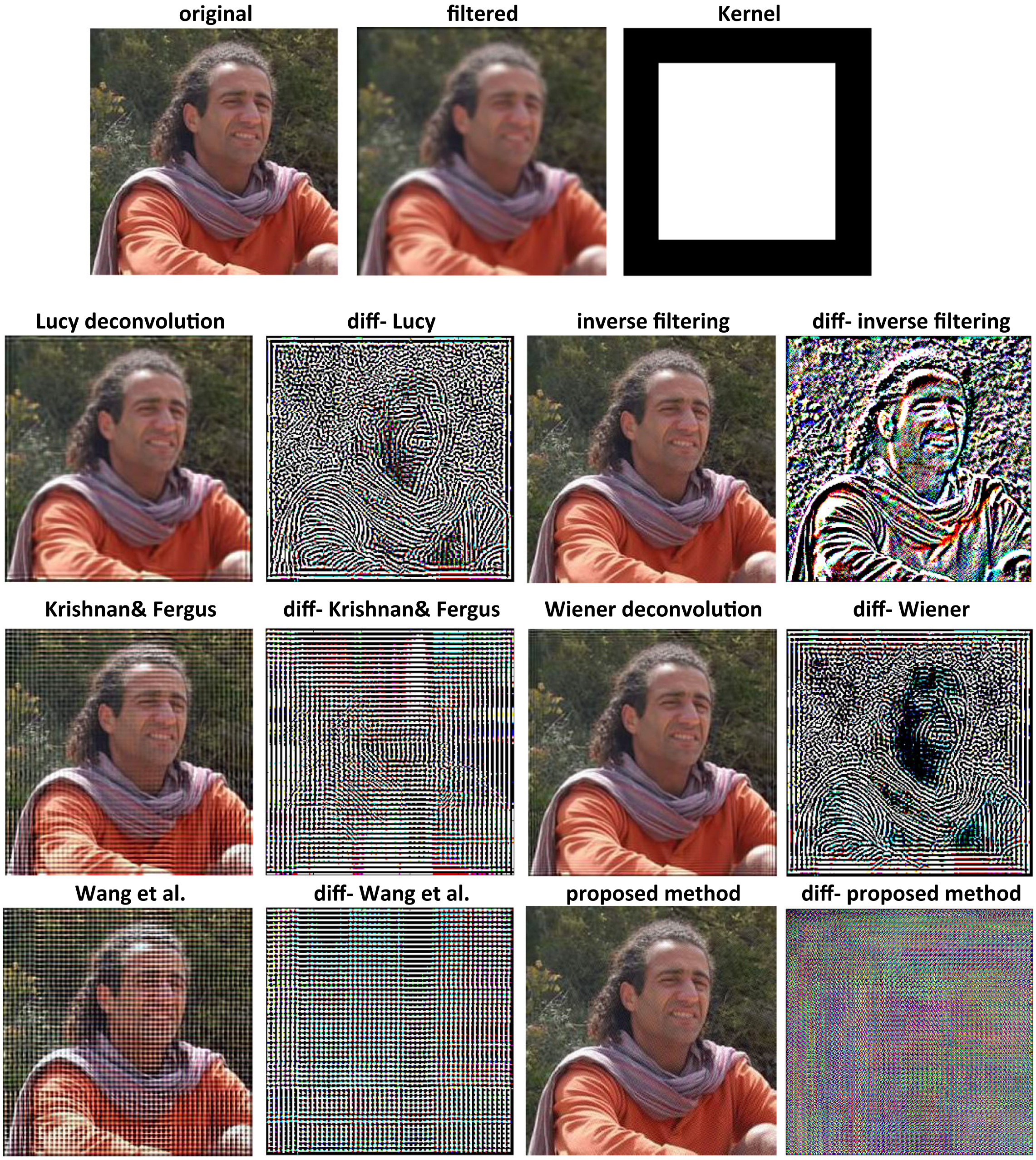
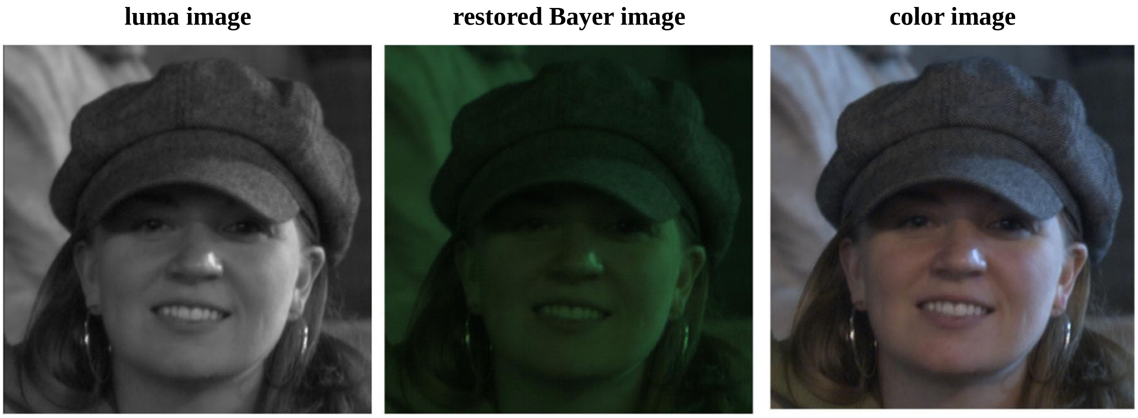


Fig. 12 Comparison of the reverse filtering to inverse filtering and to deconvolution methods with additive Gaussian noise ($\sigma = 0.01$) after the filtering. The first row shows the original image (left), filtered image using the proposed scheme (middle) and the [5×5] average kernel. The rest of the rows show pairs of restored and diff images (color limits [$-20, 20$]). RMSE: Inverse filtering (23.7), Lucy (17.5), Wiener (15.4), Krishnan–Fergus (30.0), Wang (57.8), Proposed (12.5)

the values by the division, the original image should be multiplied by 16 before the filtering. Using this approach, the Bayer/color image can be computed from the luma image after finishing the computer vision computation and save the extra memory required for storing the Bayer/RGB image. Figure 13 demonstrates the restoration of the original Bayer image from the luma channel image.

6 Summary and Conclusions

In the current paper, we propose an exact non-blind deconvolution scheme that enables bilateral transition between an image and its smoothed version. The method can revert the filtering made by average box filters or by approximated Gaussian filters that were obtained by convolution of averaging filters. The technique can be used for many image


luma image

restored Bayer image

color image

Fig. 13 Restoration of the Bayer and color images from the luma channel image

processing applications such as image editing [1], computer graphics and web commercial services where the filtering can be reverted without the need to keep the original image. The main contribution of the current work is the theoretical solution of getting the exact original image from its filtered version even if the blur kernel has zero weights in its Fourier transform. The solution may serve as a platform for future applications to come.

References

1. Barrett, W.A., Cheney, A.S.: Object-based image editing. ACM Trans. Graph. **21**, 777–784 (2002)
2. Biggs, D.S., Andrews, M.: Acceleration of iterative image restoration algorithms. Appl. Opt. **36**(8), 1766–1775 (1997)
3. Buades, A., Coll, B., Morel, J.M.: A review of image denoising algorithms, with a new one. Multiscale Model. Simul. **4**(2), 490–530 (2005)
4. Campisi, P., Egiazarian, K.: Blind Image Deconvolution: Theory and Applications. CRC Press, Boca Raton (2016)
5. Caruso, R.D., Postel, G.C.: Image editing with Adobe Photoshop 6.0. Radiographics. **22**(4), 993–1002 (2002)
6. Chan, S.H., Gill, P.E., Nguyen, T.Q.: User guide for deconvtv (matlab version 1.0). http://scholar.harvard.edu/files/stanleychan/files/deconvtv_v1_0.zip (2013). Accessed 17 July 2017
7. Chan, S.H., Khoshabeh, R., Gibson, K.B., Gill, P.E., Nguyen, T.Q.: An augmented Lagrangian method for total variation video restoration. IEEE Trans. Image Process. **20**(11), 3097–3111 (2011)
8. Donoho, D.L.: Nonlinear solution of linear inverse problems by wavelet–vaguelette decomposition. Appl. Comput. Harmon. Anal. **2**(2), 101–126 (1995)
9. Geman, D., Reynolds, G.: Constrained restoration and the recovery of discontinuities. IEEE Trans. Pattern Anal. Mach. Intell. **14**(3), 367–383 (1992)
10. Geman, D., Yang, C.: Nonlinear image recovery with half-quadratic regularization. IEEE Trans. Image Process. **4**(7), 932–946 (1995)
11. Hillery, A.D., Chin, R.T.: Iterative wiener filters for image restoration. IEEE Trans. Signal Process. **39**(8), 1892–1899 (1991)
12. Joshi, N., Zitnick, C.L., Szeliski, R., Kriegman, D.J.: Image deblurring and denoising using color priors. In: IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009, pp. 1550–1557. IEEE (2009)
13. Krishnan, D., Fergus, R.: Fast image deconvolution using hyper-Laplacian priors. In: Proceedings of the 22nd International Conference on Neural Information Processing Systems, pp. 1033–1041. Curran Associates Inc., (2009)
14. Levin, A., Fergus, R., Durand, F., Freeman, W.T.: Image and depth from a conventional camera with a coded aperture. ACM Trans. Graph. **26**(3), 70 (2007)
15. Lim, J.S.: Two-dimensional signal and image processing, vol. 1, p. 710. Prentice Hall, Englewood Cliffs (1990)
16. Michel, J.B., Shen, Y.K., Aiden, A.P., Veres, A., Gray, M.K., Pickett, J.P., Hoiberg, D., Clancy, D., Norvig, P., Orwant, J.: The Google books team: quantitative Danalysis of culture using millions of digitized books. Science **331**, 176–182 (2011)
17. Neelamani, R., Choi, H., Baraniuk, R.: Forward: Fourier-wavelet regularized deconvolution for ill-conditioned systems. IEEE Trans. Signal Process. **52**(2), 418–433 (2004)
18. Rice, J.: Mathematical Statistics and Data Analysis. Cengage Learning, Boston (2006)
19. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D **60**(1–4), 259–268 (1992)
20. Shao, L., Yan, R., Li, X., Liu, Y.: From heuristic optimization to dictionary learning: a review and comprehensive comparison of image denoising algorithms. IEEE Trans. Cybern. **44**(7), 1001–1013 (2014)
21. Stewart, C.V.: Robust parameter estimation in computer vision. SIAM Rev. **41**(3), 513–537 (1999)
22. Wang, Y., Yang, J., Yin, W., Zhang, Y.: A new alternating minimization algorithm for total variation image reconstruction. SIAM J. Imaging Sci. **1**(3), 248–272 (2008)
23. Whitt, P.: Beginning Photo Retouching and Restoration Using GIMP. Apress, New York (2014)

Yechiel Lamash was born in Haifa, Israel, in 1978. He received the B.Sc. and M.Sc. degrees in biomedical engineering from the Technion—Israel Institute of Technology, in 2007 and 2009, respectively, and Ph.D. in the field of medical image analysis from the Technion—Israel Institute of Technology, in 2014. In 2007, he joined the Workflow, Cardiac and Image Processing (WCIP) laboratory in Philips healthcare working on high-risk research projects for optional innovative applications in computed tomography (CT) imaging. During a period of seven years, he worked on various research projects that some of them were implemented to Philips Healthcare products. In 2014, he joined Samsung, Israel, as a Sr. algorithm engineer working on image processing and computer vision projects for real-time applications. In 2017, Yechiel joined the computational radiology laboratory of Boston Children's Hospital affiliated with Harvard Medical School as a research fellow.