International Journal of scientific research and management (IJSRM)
||Volume||2||Issue||10||Pages||1552-1559||2014||
Website: www.ijsrm.in ISSN (e): 2321-3418

Iterative Video De-blurring Algorithm Utilizing a Neighborhood of Unblurred Frames

Raja Sekhar Kuruba[1], Satya Prakash V.N.V[2]

[1]Rajeev Gandhi Memorial College of Engineering and Technology (Autonomous),
Affiliated to JNTU Anantapur, Nandyal, Kurnool District, Andhra Pradesh, India
razz472@gmail.com

[2]Rajeev Gandhi Memorial College of Engineering and Technology (Autonomous),
Affiliated to JNTU Anantapur, Nandyal, Kurnool District, Andhra Pradesh, India
prakashvnv@gmail.com

Abstract: In this proposed work, video deblurring can be done by iterative operations on blurred frames using Accurate Blur Kernel estimation and residual deconvolution processes. In general, while recording a video sequence using a digital camera or a digital camcorder, blurred frames may happen sparsely. The proposed work involves a novel motion deblurring algorithm in which a blurred frame can be reconstructed utilizing the high-resolution information of adjacent unblurred frames. First, a motion- compensated predictor for the blurred frame is derived from its neighboring unblurred frame via specific motion estimation. Then, an accurate blur kernel is computed using both predictor and the blurred frame. Next, a residual deconvolution is applied to both of those frames in order to reduce the ringing artifacts inherently caused by conventional deconvolution. The blur kernel estimation and deconvolution processes are iteratively performed for the deblurred frame. Simulation results show that the proposed algorithm provides superior deblurring results over conventional deblurring algorithms while preserving details.

Keywords: Blur kernel, deblurring, motion compensation, residue deconvolution, unblurred frame.

1. Introduction

A motion blur is a common artifact that causes visually annoying blurry images due to inevitable information loss. This is due to the nature of imaging sensors, which accumulate incoming light for a certain amount of time to produce an image. During the exposure time, if the camera sensor moves, motion blurred images will be obtained. Especially, such a motion blur phenomenon often occurs in a dim lighting environment where a long exposure time is required. If the motion blur is shift-invariant, it can be modeled as the convolution of a latent image I with a motion blur kernel K, i.e., a point spread function (PSF), in which the kernel describes the trace of the image sensor.

$$B = I \otimes K$$

Where B is an input blurred image and $\otimes$ is the convolution operator.

The main goal of deblurring is to reconstruct the latent image I from the input blurred image B. In general, image deblurring can be categorized into two types: blind deconvolution and non-blind deconvolution. The former is more difficult because the blur kernel is normally unknown. This paper will focus on blind deconvolution. Recently, a lot of single image blind deconvolution algorithms have been developed [1]–[9]. Fergus *et al.* used a variational Bayesian method with natural image statistics to deblur a single image [2]. Their algorithm has a drawback in that a proper image patch for kernel estimation should be selected by the user in advance. So, the performance of this algorithm highly depends on the patch selection. Shan *et al.* formulated the deblurring problem as an MAP (maximum a posteriori) problem, and proposed an image noise model of high-order derivatives, and employed a local image prior in advance in order to avoid trivial solutions [3]. However, since their algorithm incurs a heavy computational cost due to the MAP estimation process, it may often fail to deblur the images with a large size of blur kernels, on general purpose CPUs. Furthermore, Shan's algorithm tends to provide inaccurate and unstable performance for a large size of blur kernels, even though it shows satisfactory deblurring results in the case of relatively small Ks, e.g., 30 × 30 pixels or fewer. In order to mitigate the computational burden of the above-mentioned algorithms, Cho and Lee presented a fast deconvolution technique that reduces the computational overhead for latent image estimation and kernel estimation [4]. For kernel estimation, they formulated an optimization function using image derivatives and accelerated the numerical process by reducing the number of Fourier transforms needed for a conjugate gradient method. But, in spite of the fast processing of Cho's algorithm, its deblurring performance is slightly inferior to that found in the previous works. Xu and Jia proposed a two-phase kernel estimation algorithm to separate computationally expensive non-convex optimization from quick kernel initialization [5]. For precise kernel estimation, they employed spatial priors and iterative support detection kernel refinement, both of which avoid hard-thresholding the kernel elements to enforce sparsity. However, their motion deblurring schemes may fail if considerably strong and complex textures exist in the latent image. Levin *et al.* summarized the afore-mentioned single image deblurring problems in detail and provided theoretical analysis in [9].

On the other hand, the video sequences acquired by digital camcorders under a dim lighting environment may include sparsely existing blurred frames, which cause a sort of flickering phenomenon during real-time display. So, we thought that if we could exploit adjacent unblurred frames to recover the blurred frames, we could obtain much more accurate kernels and almost artifact-free images. Cho *et al.*

proposed an example-based approach which directly exploits similar patches from a nearby sharp frame of the current blurred frame [26]. But, they did not incorporate the de-convolution techniques into the proposed patch synthesis algorithm.

This paper presents a video deblurring algorithm that utilizes the neighborhood of unblurred frames. First, an initial blur kernel of an input blurred frame is estimated, and the nearest neighbor of the blurred frame is chosen among the unblurred frames. And then, the selected frame is deliberately blurred using the initially estimated kernel. Next, motion estimation is performed between the input blurred frame and the artificially blurred neighbor, and the motion compensated predictor of the blurred input is obtained. Then, the blur kernel is refined using the input blurred frame and its motion-compensated predictor, and deconvolution based on the refined blur kernel is finally applied to the blurred frame. Here, in order to mitigate the so-called ringing phenomenon during deconvolution, we propose a residual deconvolution that deconvolutes the motion-compensated residue. This entire process is iterated until the residue value becomes sufficiently small. From the intensive simulation results, we were able to find that the proposed algorithm provides significantly better visual quality, and it also shows about 4dB higher ISNR (the increase in signal to noise ratio) at maximum than the state-of- the-art deblurring methods.

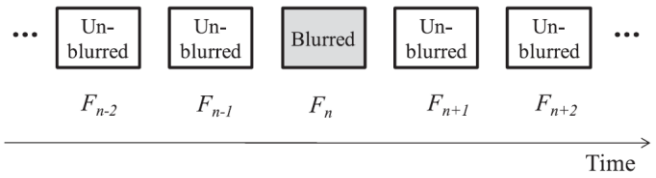

Figure 1: Example of a Blurred Frame in a Video Sequence

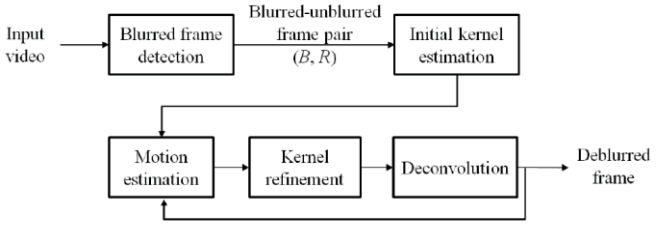

Figure 2: Block Diagram of the Proposed Algorithm

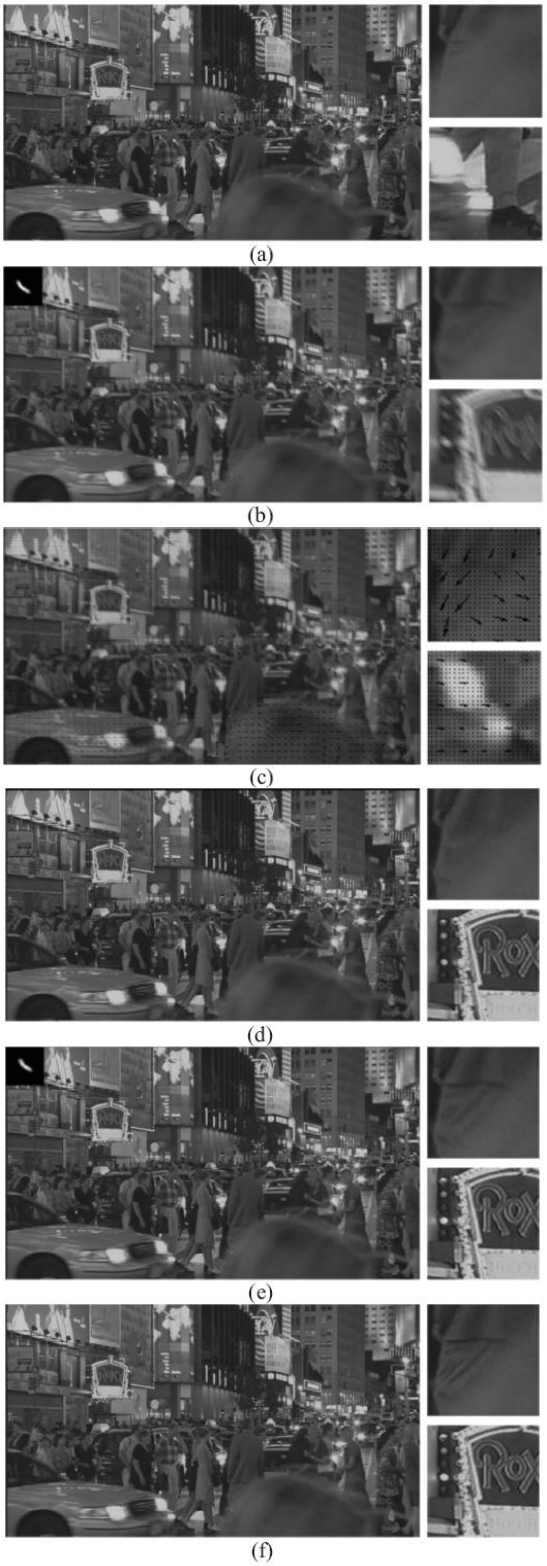

Figure 3: Feasibility of the proposed deblurring scheme. (a) Reference unblurred frame *R*. (b) Blurred input *B*. (c) Motion vector field on *B*. (d) Motion-compensated frame I_m. (e) Deblurred frame. (f) Original frame. Here, the upper-left boxes in (b) and (e) indicate the original and estimated blur kernels, respectively.

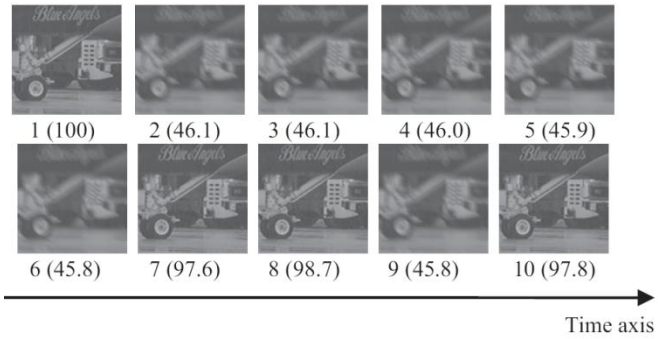

Figure 4: Example of blur detection. The leftmost tile indicates the representative tile in the first frame, and the others are the tracked tiles. The bottom numbers stand for frame numbers and the figures in parentheses refer to relative edge energy.

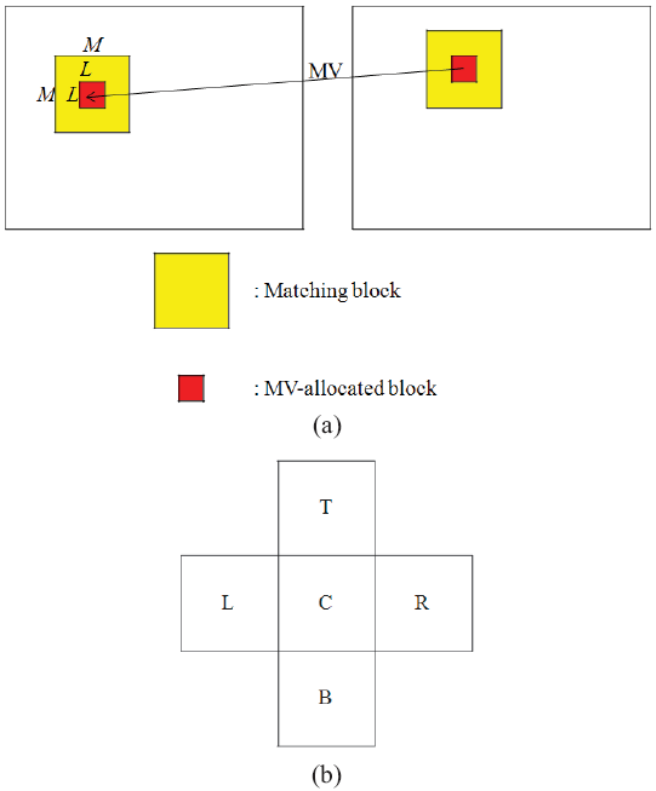

Figure 5: (a) Motion estimation. (b) Four neighbor blocks for OBMC.

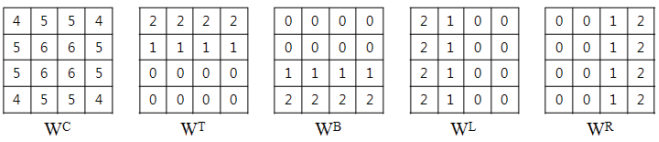

Figure 6: Weight matrices for OBMC.

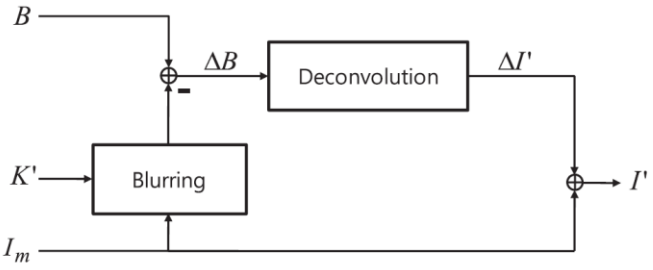

Figure 7: Proposed residue deconvolution process.

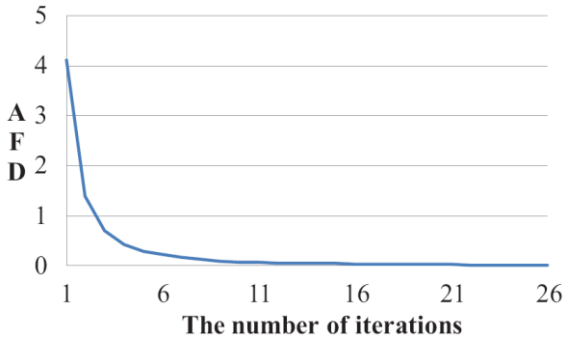

Figure 8: Average frame difference between the current deblurred frame and the resulting frame of the previous iteration according to the number of iterations.

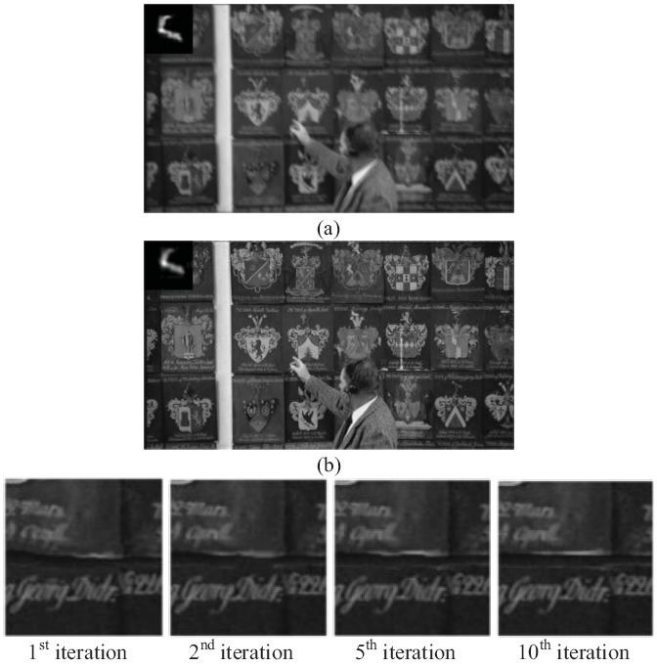

Figure 9: (a) Blurred frame and the original blur kernel. (b) Our deblurring result and the estimated blur kernel.

2. Proposed algorithm

This paper assumes that the blur kernel is shift-invariant, and that the blur phenomenon sparsely happens in a video sequence. Figure 1 illustrates a sparsely occurring blurred frame in a video sequence recorded by digital camcorder under a dim lighting environment. Our goal is to reconstruct a high quality version of the blurred frame in the video sequence.

Let B and R denote a target blurred frame F_n and its nearest unblurred frame F_{n-1}, respectively. Figure 2 provides an overview of the proposed algorithm. First, the blurred frame B can be detected using edge energy information. Since the edge energy of the blurred frame is noticeably small, we can find the position of each blurred frame by comparing edge energies. Next, an initial kernel of the blurred frame is estimated and R is deliberately blurred using the initial blur kernel. Then, motion estimation is performed between B and the artificially blurred R. Next, the blur kernel is re-computed using B and the motion-compensated predictor of B, which is denoted by I_m. Finally, once K is refined, we can use Eq. (1) to non-blindly recover I from B. As can be seen in Figure 2, this entire process is iterated until an acceptable I is produced.

The main contribution of the proposed algorithm is to present an effective deblurring mechanism for video sequences. First, accurate registration between two adjacent frames in a video sequence, i.e., B and R, accelerates the precise estimation of blur kernel. Also, subsequent residual deconvolution significantly suppresses visually annoying ringing artifacts in the deblurred video frames. For example, Figure 3 shows that the deblurring result derived from I_m provides significantly better visual quality than I_m. For this experiment, a specific frame in a video sequence is artificially blurred, as shown in Figure 3(b). Note that even though I_m possesses most of the details of the original, it still has limitations. From the close-ups of Figure 3(d) and (e), we can observe that the proposed algorithm successfully reconstructs even fine details in motion [see Figure 3(c) and (e)]. The following subsections describe the key components of the proposed algorithm.

1.1. Initial kernel estimation and registration between blurred and unblurred frames

In this step, we derive an initial blur kernel of B and produce the motion-compensated predictor I_m corresponding to B. Prior to the main processing, the blurred frames should be detected, as in Figure 2. Assume that the blur detection is performed on a shot basis in the video sequence, and the first frame of each shot is not blurred. We compute the edge energies of all the overlapping tiles in the first frame, and choose a tile having maximum edge energy as the representative tile T_R. Here, the tile size is empirically set to 128 × 128, a half of each tile is overlapped with its neighbors, and the edge energy of each pixel in the tile is computed via Sobel operator and averaged. Note that tiles having higher edge energy tend to be more sensitive to blur phenomenon. Then, for the following frames, T_R is tracked and the edge energy of the tracked tile is compared with that of T_R. In this paper, if the edge energy of the tracked tile in a certain frame drops to 30% below that of T_R, we determine that frame to be the blurred frame B, and define the nearest unblurred frame to B as R. A typical full search block matching algorithm is employed for this tracking process, and the search range is set to [−64], [64] × [−64], [64]. Figure 4 illustrates an example of the blur detection. Several frames in *Jets* sequence are artificially blurred for this example. Every edge-energy is normalized by re-scaling the edge energy of T_R to 100. We can observe that the artificially blurred frames, i.e., the 2*nd*, 3*rd*, 4*th*, 5*th*, and 9*th* frames are exactly chosen as the blurred frames according to the afore-mentioned rule.

Next, for each B we generate a sharp reference image that is motion-compensated from R by the motion vector (MV) field between B and R. In general, it is very hard to find an accurate MV field between blurred and unblurred frames. Yuan *et al*. presented an effective alignment approach for a blur kernel. But, their kernel estimation highly depends on the search intervals and ranges for the rotation angle and the scale, and rarely handles more complex motion, e.g., perspective transformation or non-global motion. Thus, we estimate an initial blur kernel for B and artificially blur R by using the estimated blur kernel K_i. For the kernel estimation, we employed a fast kernel estimation algorithm proposed in [4]. Then, motion estimation (ME) is applied between B and the artificially blurred R, i.e., I_b. In order to minimize the motion compensated error and artifacts, we adopted a hierarchical ME from [25], which was used for motion-compensated super resolution.

First, the MV for an overlapping $M \times M$ matching block is searched by a full search, and the selected MV for the $M \times M$ block is allocated to its central $L \times L$ block, as is shown in Figure 5(a). Direct motion compensation often causes blocking artifacts. In order to reduce such artifacts, we adopt the so-called OBMC (overlapped block motion compensation), exploiting the four neighboring MVs, as shown in Figure 5(b). In this paper, the OBMC is performed on a 4 × 4 block basis. The pixel value at (i, j) of a certain 4 × 4 block is motion-compensated as follows:

$$\begin{aligned} I_m(i,j) = {} & W^C(i,j)\cdot R^C(i,j) + W^T(i,j)\cdot R^T(i,j) \\ & + W^B(i,j)\cdot R^B(i,j) + W^L(i,j)\cdot R^L(i,j) \\ & + W^R(i,j)\cdot R^R(i,j) \end{aligned} \quad (2)$$

where the weight matrices for the OBMC are defined in Figure 6. In Eq. (2), $R^C(i, j)$, $R^T(i, j)$, $R^B(i, j)$, $R^L(i, j)$, $R^R(i, j)$ are the pixel values at (i, j) in the motion-compensated blocks corresponding to current MV, MVs of the top neighbor block, the bottom neighbor block, the left neighbor block, and the right neighbor block, respectively. Note that ME is performed between B and I_b, but the motion-compensated blocks are derived from the unblurred reference frame R. So, we can obtain a sharp reference frame I_m for the deblurring of B.

The accuracy of the initial blur kernel and the high quality of I_m which can be achieved in this step quite positively affect the overall performance of the proposed algorithm. Since an inaccurate blur kernel causes poor registration, it deteriorates I_m. As a result, the degraded I_m gives rise to an unsatisfactory visual quality of the final deblurred frame. Even though further steps such as kernel refinement and deconvolution mitigate the performance degradation caused by inaccurate blur kernel, the performance decline of the overall system is ultimately unavoidable.

1.2. Kernel Refinement

Now, we refine the blur kernel by using I_m and B. Here, the initial blur kernel is K_i . Note that as I_m is closer to the original latent frame, the refined kernel may be more similar to its original kernel. Eq. (1) can be represented in matrix form as follows:

$$\mathbf{b} = \mathbf{A}\,\mathbf{k} \quad (3)$$

where **b**, **A**, and **k** denote matrix forms of B, I, and K, respectively. Let K' be the refined kernel. We can derive the best K' via a minimization process of Eq. (4).

$$K' = \min_k \| Ak - b \|^2 + \lambda \| k \|^2 \quad (4)$$

Here, Tikhonov regularization is employed to find a stable solution, and λ is empirically set to 5. In order to solve Eq. (4), we use the so-called conjugate gradient (CG) method. Then, the gradient of the cost is defined by

$$\frac{\partial}{\partial k}[\| Ak - b \|^2 + \lambda \| k \|^2] = 2A^T Ak + 2\lambda k - 2A^T b \quad (5)$$

Eq. (5) should be evaluated many times in the minimization process. So, this direct computation of matrix operations requires heavy computational and storage overhead. Fortunately, since $\mathbf{A^T Ak}$ and $\mathbf{A^T b}$ correspond to convolution, we can accelerate the computation by fast Fourier Transforms (FFTs), as in Eq. (7).

$$A^T Ak = F^{-1}[\overline{F(1)} \circ F(1) \circ F(K)] \quad (6)$$

$$A^T b = F^{-1}[\overline{F(1)} \circ F(B)] \quad (7)$$

Here, *F(X)* and $\overline{F(X)}$ indicate the FFTs of a matrix *X* and its complex conjugate, respectively. Also, ° stands for pixel-wise multiplication. In addition, we adopted a coarse-to-fine approach to reduce computational complexity. The pyramids are constructed using a down sampling factor of $1/\sqrt{2}$ until the kernel size at the coarsest level reaches 9 × 9. Finally, the refined kernel *K'* is normalized for energy preservation.

1.3. Deconvolution

Given the refined blur kernel K', the final deblurred frame I' can be reconstructed from I_m and *B*, as shown in Figure 7. In this paper, we start from the concept of residual deconvolution proposed by Yuan *et al.* in [13]. Instead of the direct deconvolution of *B*, Yuan *et al.* deconvoluted the residual blurred image to remove the ringing artifacts. However, since there is generally a shift between the neighboring frame pairs, such a deconvolution rarely works for video sequences. So, we utilize I_m, i.e., the motion-compensated predictor from *R* instead of *R* itself for deconvolution. Thus, we perform deconvolution on the residual blurred frame $\Delta B \equiv \Delta I \otimes K'$ to recover the motion-compensated residual frame ΔI. ΔB is derived from the following equation:

$$\Delta B = B - (I_m \otimes K') \quad (8)$$

From Eq. (8), $\Delta I'$ can be derived via deconvolution. Here, we employed a simple deconvolution algorithm using a Gaussian-prior. Finally, the deblurred frame is obtained by $I' = I_m + \Delta I'$. Note that as ME becomes more accurate, ΔB has less energy. Therefore, while preserving sharp edges thanks to I_m, the residual deconvolution can predict $\Delta I'$ with noticeably suppressed ringing artifacts.

For more accurate deblurring, the above-mentioned processes are iterated as shown in Figure 2. Note that at the 1*st* iteration, ME is applied to I_b and *B*, but from the next iterations it is performed directly between I' and *R* to find accurate MVs. The iteration is terminated if the average frame difference between the result frames of the current and previous iterations (AFD) is sufficiently small. Figure 8 shows the AFD values per pixel according to the number of iterations.

3. Experimental results

For quantitative evaluation, we applied the proposed algorithm to several well-known 1280 × 720 video sequences that included artificially blurred frames. For this experiment, we blurred several frames of four well-known 1280 × 720 video sequences: *Shields*, *Night*, *City*, and *Jets* by using a few blur kernels. For intensive experiments, we selected video sequences having various motion types. For instance, the dominant motion of *Shields* is panning,and *Night* includes many different object motions. Also, *City* has a little rotation as well as panning, and *Jets'*s dominant motion is zooming. For motion estimation, the matching block size and motion search range were set to 16 × 16 and ±64, respectively. The MV for the overlapping 16 × 16 matching block is searched by the full search, and the selected MV for the 16 × 16 block is allocated to its central 4 × 4 block, as shown in Figure 5(a). We compared the proposed algorithm with four state-of-the- art algorithms: Fergus's [2], Shan's [3], Cho's [4], and Xu's algorithms [5]. For fair comparison, we employed the so-called ISNR (the increase in signal to noise ratio) proposed by Almeida [6]. The ISNR is defined as follows;

$$ISNR = 10 \log_{10} \frac{\| f(B) - I \|^2_2}{\| f(I') - I \|^2_2} \quad (9)$$

where *f* (.) indicates the affine transformation, and the best parameters are estimated between two corresponding frames prior to ISNR calculation. The MATLAB routines for computing the ISNR are available at http://www.lx.it.pt/~mscla/BID_QM.htm. The higher ISNR values indicate that I' becomes closer to *I*. First, we compared the ISNR values of the proposed algorithm according to the frame distances between *B* and *R* because the blur phenomenon happens in a burst, and the distance of *R* from *B* may be long in such a case. Table 1 shows the ISNR comparison results.

Table 1:

ISNR comparison according to the frame distances between B and R [dB]

	The Frame Distance Between B and R				
	1	2	3	4	5
Shields	7.63	7.01	6.18	6.39	6.96
Night	9.82	8.94	8.09	8.79	6.01
City	9.03	9.24	9.13	9.17	8.77
Jets	5.38	6.26	6.09	5.56	5.59

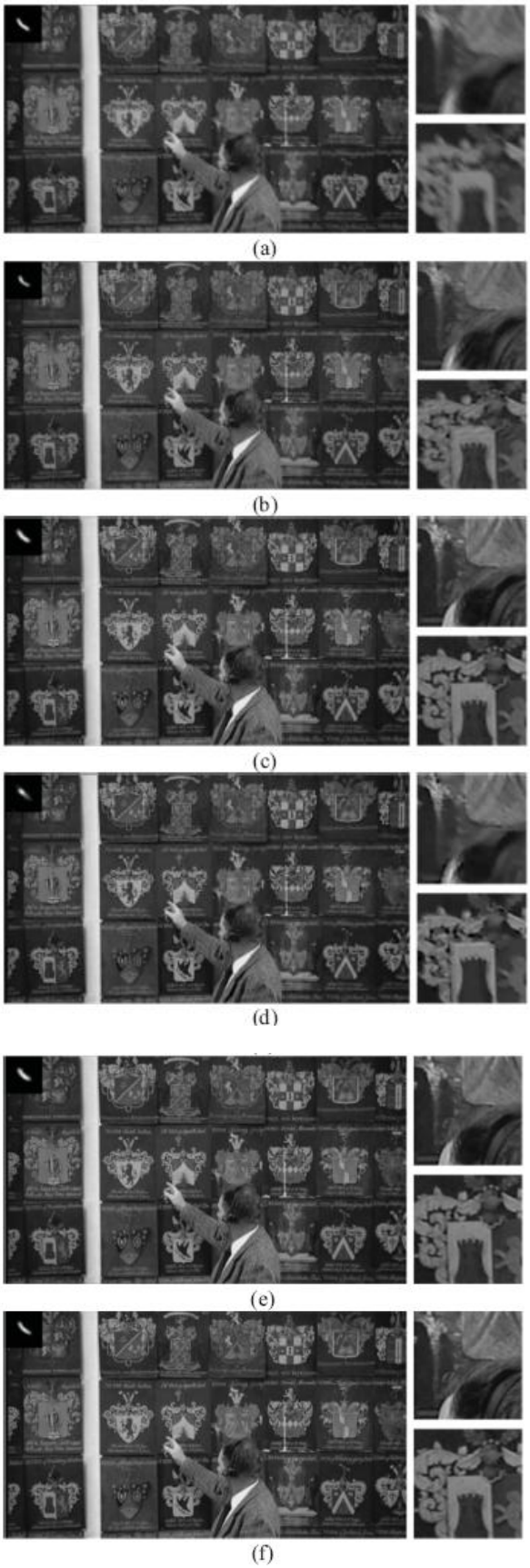

Figure 10: Deblurred results for the *Shields* sequence. (a) Input blurred frame. (b) Fergus's. (c) Shan's. (d) Cho's. (e) Xu's. (f) Proposed algorithm. Here, the upper-left boxes indicate original or estimated blur kernels.

Figure 11: Deblurred results for the *Night* sequence. (a) Input blurred frame. (b) Fergus's. (c) Shan's. (d) Cho's. (e) Xu's. (f) Proposed algorithm. Here, the upper-left boxes indicate original or estimated blur kernels.

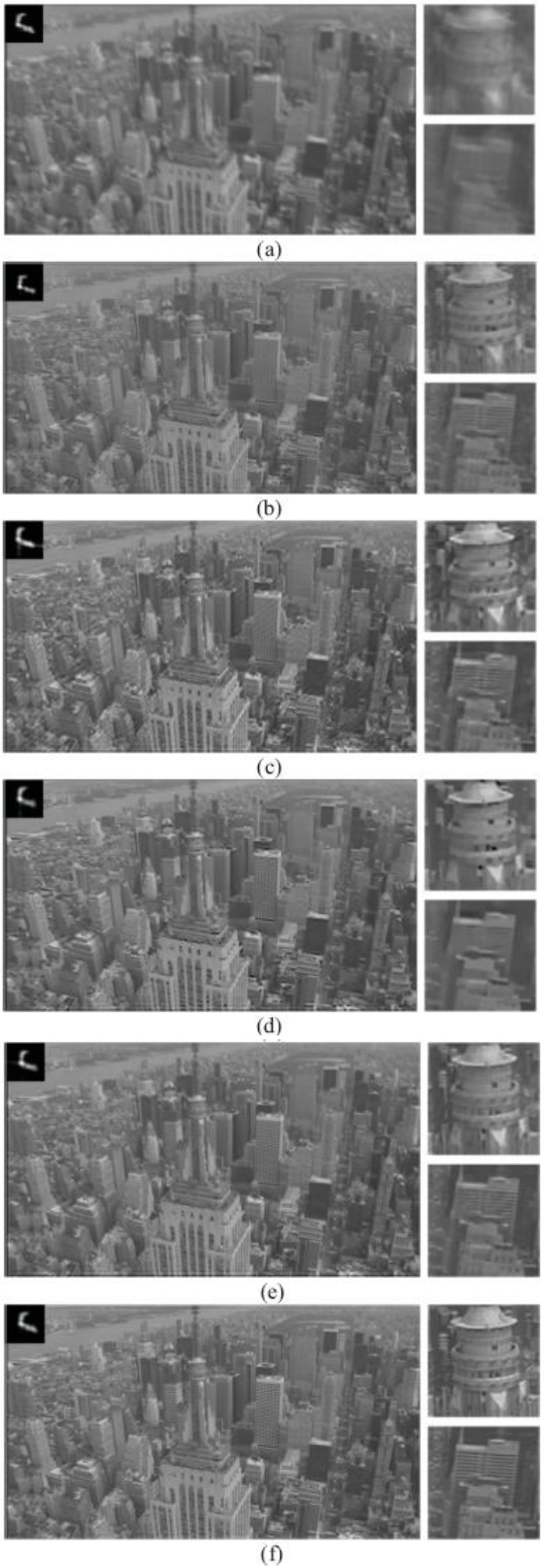

Figure 12: Deblurred results for the *City* sequence. (a) Input blurred frame. (b) Fergus's. (c) Shan's. (d) Cho's. (e) Xu's. (f) Proposed algorithm. Here, the upper-left boxes indicate original or estimated blur kernels.

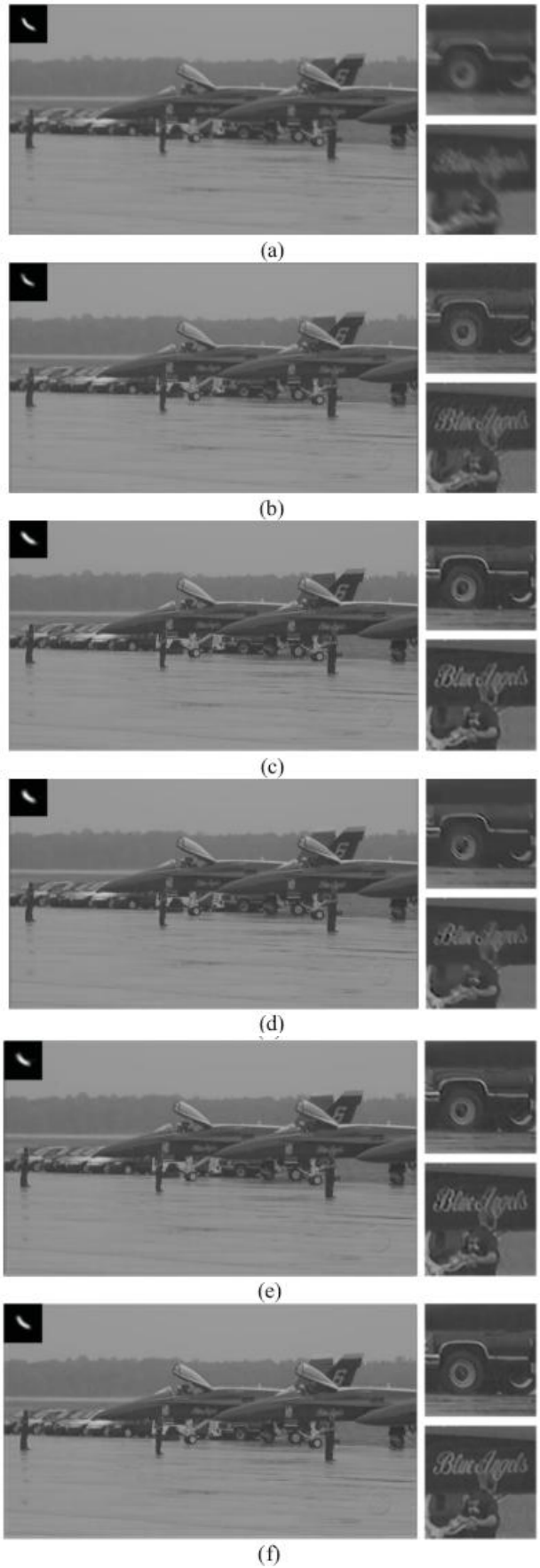

Figure 13: Deblurred results for the *Jets* sequence. (a) Input blurred frame. (b) Fergus's. (c) Shan's. (d) Cho's. (e) Xu's. (f) Proposed algorithm. Here, the upper-left boxes indicate original or estimated blur kernels.

4. Conclusion

This paper presents an iterative video deblurring algorithm utilizing a neighborhood of unblurred frames. First, the sharp predictor of a blurred frame is derived from its neighboring unblurred frame using motion estimation. Second, an accurate blur kernel is computed using the predictor and the blurred frame. Third, again using both of those frames, a residual deconvolution is proposed to significantly reduce the ringing artifacts inherent in conventional deconvolution. From experimental results we proved that the proposed algorithm reconstructs details better than conventional algorithms do, with fewer ringing artifacts. In this paper, we assume that the blur kernel is shift-invariant. As for further work, we plan to extend our approach to shift-variant blur kernels.

References

[1] D. Kundur and D. Hatzinakos, "Blind image deconvolution," *IEEE Signal Process. Mag.*, vol. 13, no. 3, pp. 43–64, May 1996.

[2] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, "Removing camera shake from a single photograph," in *Proc. ACM SIGGAPH*, 2006, pp. 787–794.

[3] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," *ACM Trans. Graph.*, vol. 27, no. 3, p. 73, Aug. 2008.

[4] S. Cho and S. Lee, "Fast motion deblurring," *ACM Trans. Graph.*, vol. 28, no. 5, p. 145, Dec. 2009.

[5] L. Xu and J. Jia, "Two-phase kernel estimation for robust motion deblurring," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 157–170.

[6] M. S. C. Almeida and L. B. Almeida, "Blind and semi-blind deblurring of natural images," *IEEE Trans. Image Process.*, vol. 19, no. 1, pp. 36–52, Jan. 2010.

[7] J. F. Cai, H. Ji, C. Liu, and Z. Shen, "Blind motion deblurring from a single image using sparse approximation," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 104–111.

[8] J. Jia, "Single image motion deblurring using transparency," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.

[9] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Understanding blind deconvolution algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2354–2367, Dec. 2011.

[10] A. Rav-Acha and S. Peleg, "Two motion-blurred images are better than one," *Pattern Recognit. Lett.*, vol. 26, no. 3, pp. 311–317, Feb. 2005.

[11] J. Chen, L. Yuan, C.-K. Tang, and L. Quan, "Robust dual motion deblurring," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[12] J. F. Cai, H. Ji, C. Liu, and Z. Shen, "High-quality curvelet-based motion deblurring from an image pair," in *Proc. Conf. Comput. Vis. Pattern*

[13] L. Yuan, J. Sun, L. Quan, and H. Y. Shum, "Image deblurring with blurred/noisy image pairs," *ACM Trans. Graph.*, vol. 26, no. 3, p. 1, Jul. 2007.

[14] P. Huang, Y. Lin, and S. Lai, "Image deblurring with blur kernel estimation from a reference image patch," in *Proc. Int. Conf. Pattern Recognit.*, 2008, pp. 1–4.

[15] S. Cho, Y. Matsushita, and S. Lee, "Removing non-uniform motion blur from images," in *Proc. Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

[16] M. Ben-Ezra and S. K. Nayar, "Motion-based motion deblurring," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 689–698, Jun. 2004.

[17] A. Agrawal, Y. Xu, and R. Raskar, "Invertible motion blur in video," *ACM Trans. Graph.*, vol. 28, no. 3, p. 95, Aug. 2009.

[18] B. Bascle, A. Blake, and A. Zisserman, "Motion deblurring and superresolution from an image sequence," in *Proc. Eur. Conf. Comput. Vis.*, 1996, pp. 571–582.

[19] Y. Huang and N. Fan, "Inter-frame information transfer via projection onto convex set for video deblurring," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 2, pp. 275–284, Apr. 2011.

Author Profile

Raja Sekhar Kuruba received B.Tech degree in Electronics and Communication Engineering from Sri Sai Institute of Technology and Science(Affiliated to JNTU Anantapur) in 2012. Now, he is pursuing M.Tech degree in Digital Systems and Computer Electronics from Rajeev Gandhi Memorial College of Engineering and Technology(Autonomous) from 2012 to till date.

Satya Prakash V.N.V Working as Associate Professor in the department of Electronics and Communications Engineering at Rajeev Gandhi Memorial College of Engineering and Technology, Nandyal(Kurnool district), Andhra Pradesh, India.