ISSN: 2321-7782 (Online)
Volume 2, Issue 9, September 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Image and Video Deblurring Algorithm Using Normalized Sparsity Measure

Rosy John[1]
Department of Electronics and Communication Engineering
Amal Jyothi College of Engineering
Kanjirappally – India

Ajai Mathew[2]
Department of Electronics and Communication Engineering
Amal Jyothi College of Engineering
Kanjirappally – India

***Abstract:** Motion blur is an artifact that causes visually annoying images. In this work a novel algorithm to deblur blurred images and video frames is proposed. Prior to deblurring it automatically checks whether an image or frame is blurred or not using a method based on Cumulative Probability of Blur detection. If the image is blurred, deblurring algorithm is applied obtain the true image. This method utilizes normalized sparsity measure to recover the sharp image. The algorithm is simple fast and robust. Experimental results show that this method can effectively remove motion blur and recover sharp image.*

***Keywords:** motion blur; deblurring; cumulative probability; normalized sparsity measure; blind image deconvolution.*

I. Introduction

Blur is a result of imperfect image formation process. It may be either due to the relative motion between the camera and the object or due to an out of focus optical system. Based on this blur is classified into motion blur and out of focus blur. Motion blur mainly occurs in a dim lighting environment where long exposure time is required. Motion blur can be modelled as the convolution of the sharp image u with the blur kernel k or point spread function (PSF). PSF refers to the extent to which an image of a point source is blurred by the motion blur.

$$g = u \otimes k + n \qquad (1)$$

where g is the blurred image $\otimes$ is the convolution operator k is the blur kernel and n is the noise. The goal of deblurring is to recover sharp image from the input blurred image. Generally deblurring algorithms are classified into two blind image deconvolution and non blind image deconvolution. In blind image deconvolution blur kernel is known whereas in non blind image deconvolution blur kernel is unknown hence it is more difficult to solve.

II. Related Works

Recovering high quality image from a blurred image is a well studied problem. However constructing true image from a blurred image without artifacts is still a challenging problem. Fergus *et al* introduced a method for removing camera shake effects from photographs [1]. Their efforts were focused on kernel estimation and the estimated kernels seem to match the camera motion. But the recovered image often contains ringing artifacts. Shan *et al.* proposed a deblurring algorithm that suppress ringing artifact using Maximum a posterior approach (MAP) [2]. However, their method fails to deblur images with large size kernels since their algorithm incurs heavy computational cost. A fast motion deblurring approach using image derivatives was proposed by Cho and Lee [3]. They accelerated the sharp image estimation process by introducing a prediction step in the iterative deblurring process. Even though the algorithm ensures a fast processing its deblurring performance is slightly inferior to the previous methods. Ben Ezra and Nayar developed a hybrid camera design that uses a fundamental tradeoff between spatial and temporal resolution to obtain the camera motion information [4]. This information was used to obtain the Point Spread Function. Since it is a hardware approach it cannot be applied to general purpose video cameras.

This paper proposes an image and video deblurring algorithm using normalized sparsity measure. Before main processing it is checked that whether an image is blurred or not. If an image is blurred, blur kernel is estimated and after blur kernel estimation sharp image is obtained by deconvolution of the blurred image with the estimated blur kernel.

III. PROPSED ALGORITHM

This paper presents a novel algorithm to deblur image and video sequences. Fig. 1 provides an overview of the proposed algorithm. Our algorithm has three main steps. First blur detection based on Cumulative Probability of Blur Detection (CPBD). Secondly bind kernel estimation using normalized sparsity measure. The last step is Image deconvolution.

The main advantage of this algorithm is that it is simple fast and robust. Here blur is detected by using CPBD sharpness metric which is based on Human visual system hence it detects blur as perceived by human eye. Normalized sparsity measure used for blur kernel estimation provides a tractable optimization algorithm. For sharp image recovery image deconvolution method based on hyper laplacian prior is used which is robust to small kernel errors.

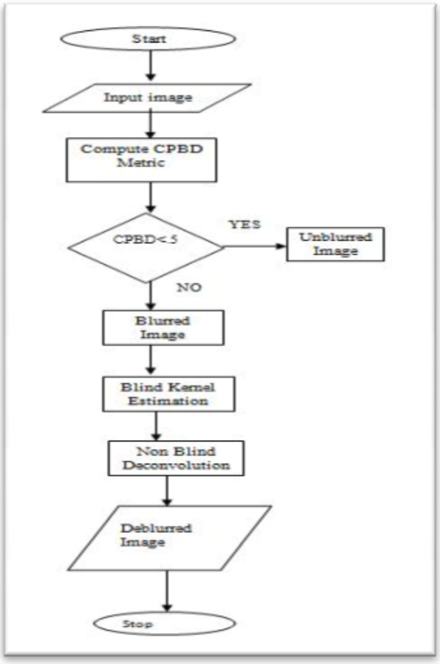

Fig. 1 Proposed Algorithm

a. Blur Detection

In this step we check whether an image is blurred on not. Blur detection is based on CPBD sharpness. The basis of CPBD is Just Noticeable Blur as proposed in [7]. Just Noticeable blur is the minimum amount of perceived blurriness given a contrast higher than Just Noticeable Difference is given by the psychometric function in equation 2

$$P_{BLUR} = P(e_i) = 1\text{-}exp(-|w(e_i) / w_{JNB}(e_i)|^{\beta}) \tag{2}$$

Where w (e_i) is the width of the edge e_i , $w_{JNB}(e_i)$ is the JNB width which depends on the local contrasts and β is least square fitting parameter. If contrast of a block is below 50 $w_{JNB}(e_i)$ is 5 and if contrast of a block is above 51 $w_{JNB}(e_i)$ is 3 [7].

First the image is divided into 64×64 blocks and then number of edge pixels in each block is calculated. If the number of edge pixels is less than 0.2% of the total number of pixels in a block it is characterized as a non edge block. Non edge blocks are not processed further. Contrast of each block is calculated and the w_{JNB} is obtained. For every edge block w (e_i) of each edge pixel is calculated. If $w(e_i) = w_{JNB}(e_i)$ then probability of blur detection P_{BLUR} = 63% = P_{JNB}. Then cumulative probability of blur detection can be calculated as

$$CPBD = P(P_{BLUR} \leq P_{JNB}) = \sum_{P_{BLUR}=0}^{P_{BLUR}=P_{JNB}} P(P_{BLUR} \tag{3}$$

P (P_{BLUR}) gives the probability density function at a given P_{BLUR} .CPBD gives the measure of percentage of edges at which blur is not detected in an image. Hence higher the metric the sharper will be the image. It is a value between 0 and 1. In this paper CPBD threshold value is set as 0.5. If CPBD metric of an image is below 0.5 it is considered to be blurred

b. Blur Kernel Estimation

Blind kernel estimation is an ill-posed problem. To solve such problem regularization technique is needed. Here we use l_1 /l_2 regularization. L_1 norm is used to impose signal sparsity. Blur decreases both l_1 and l_2 norm of images but l_2 norm is reduced more and hence the ratio of the two increases. Simply l_1 /l_2 norm can be interpreted as the normalized version of l_1 norm or normalized sparsity measurement.

The first step in kernel estimation is to obtain high frequencies of the image using discrete filters f_x= [1 -1] and f_y = $[1\ -1]^T$. The high frequency version of the image is given by y = [f_x g f_y g]. By taking into account the image formation model of equation (1) the cost function for blurring is given by

$$\min_{x,k} \quad \lambda \|x \otimes k - y\|_2^2 + \frac{\|x\|_1}{\|x\|_2} + \psi \|k\|_1 \tag{4}$$

This is a constrained optimization problem where the constraints are $k \geq 0, \sum_i k = 1$. Here k is the blur kernel, x is the unknown sharp image and $\otimes$ is the convolution operator. The second term is the $l_1 \backslash l_2$ regularization on x. l_1 regularization is added to k to reduce the noise level. λ and ψ are the parameters that controls the kernel and image regularization. The cost function formulated here is highly non-convex hence it is difficult to optimize directly. Here we set initial values for x and k and then both are updated alternately. We solve x and k update problem using Iterative shrinkage threshold algorithm (ISTA) and Iterative reweighted least squares method respectively. The cost function of (4) can be hence divided as x update problem and k update problem x Update problem: The x update sub problem is given as

$$\min x \, \lambda \|x \otimes k - y\|_2^2 + \frac{\|x\|_1}{\|x\|_2} \tag{5}$$

This is highly non convex due to the presence of $l_1 \backslash l_2$ regularizer. If we fix the denominator of (5) from the previous iteration the problem becomes

$$\min_{x} \lambda \|x \otimes k - y\|_2^2 + \|x\|_1 \tag{6}$$

Which is similar to the classical linear inverse sub problem that can be solved by ISTA [5]. Every iteration of this algorithm involves matrix vector multiplication involving K and K^T followed by a shrinkage step given by

$$v = y - tK^T \left(kx^j - y\right) \tag{7}$$

$$x^{j+1} = s_{t\lambda}(v) \tag{8}$$

Where s is the shrinkage operation on vector v, t is an appropriate step size. The value of j is varied from 0 to $N-1$ (N is the number of iterations). Shrinkage operator is given by

$$s_\alpha(x_i) = \max(|x_i| - \alpha, 0)\, sign(x_i) \tag{9}$$

The shrinkage operator shrinks each component of the input vector towards zero. The ISTA is the inner loop of the algorithm. The outer loop of the algorithm estimates the weighting term λ.

K update sub problem: The kernel update sub problem is given by

$$\min_{k} \lambda\|x\otimes k - y\|_2^2 + \Psi\|\mathrm{k}\|_1 \tag{10}$$

Subject to the constraints $k \geq 0$, $\sum_i k_i = 1$. Unconstrained IRLS is used to solve the minimization problem. It is followed by projecting *k* onto the constraints. It is done by setting the negative elements to zero and renormalizing *k.* The inner IRLS is solved by using Conjugate Gradient Iterations [6].

Multiscale implementation is used for large size kernels because it requires excessive number of x and k updates to converge to a reasonable solution. It is a coarse to fine approach. Level with size ratio √2 between them is used. The kernel size at the coarsest level is 3×3. The input blurry image is down sampled and then discrete gradients are taken to form the input *y* at each level. After kernel estimate k and sharp gradient image x are computed they are up sampled. Bilinear interpolation is used to perform rescaling operations.

c. Image Deconvolution

After kernel estimation the sharp image can be recovered by using the recovered kernel using non-blind deconvolution techniques. The simplest is Richardson-Lucy Algorithm. The disadvantage of this method is that it causes ringing artifacts in the deblurred output if there are errors in the estimated kernel. Here we use the non-blind deconvolution method of [8]. This algorithm utilizes a continuation method to solve the following cost function

$$\min \lambda\|u\otimes k - g\|_2^2 + \|f_x\, g\|_\alpha + \left\|f_y\, g\right\|_\alpha \tag{11}$$

f_x and f_y are the derivative filters mentioned in section B. The value of λ and α are chosen as 3000 and 0.5 respectively for all results. For image recovery we use l_p type regularizers as in [8].

IV. SIMULATION RESULTS

We test the algorithm on the data set provided by Dilip and Fergus [6]. The data set consists of four blurred images of size 255×255. The input blurred images and deblurred results are shown in Fig. 2 and Fig.3

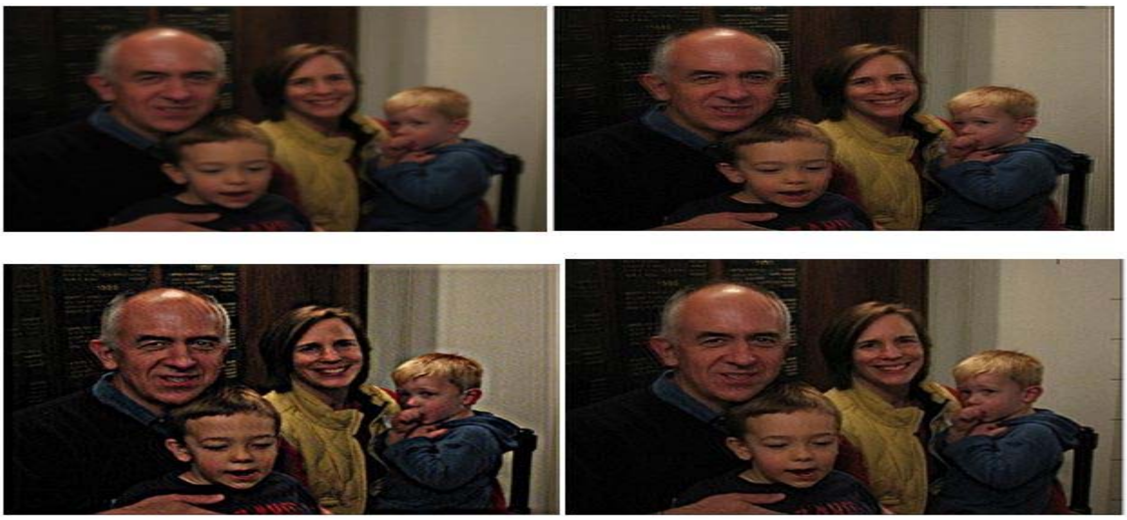

Fig.2. Deblurred Result: Top left: Input blurry image; Top right: Deblurred with algorithm of [3]; Bottom Left:Deblurred with algorithm of [9]; Bottom right: Deblurred with our Algorithm

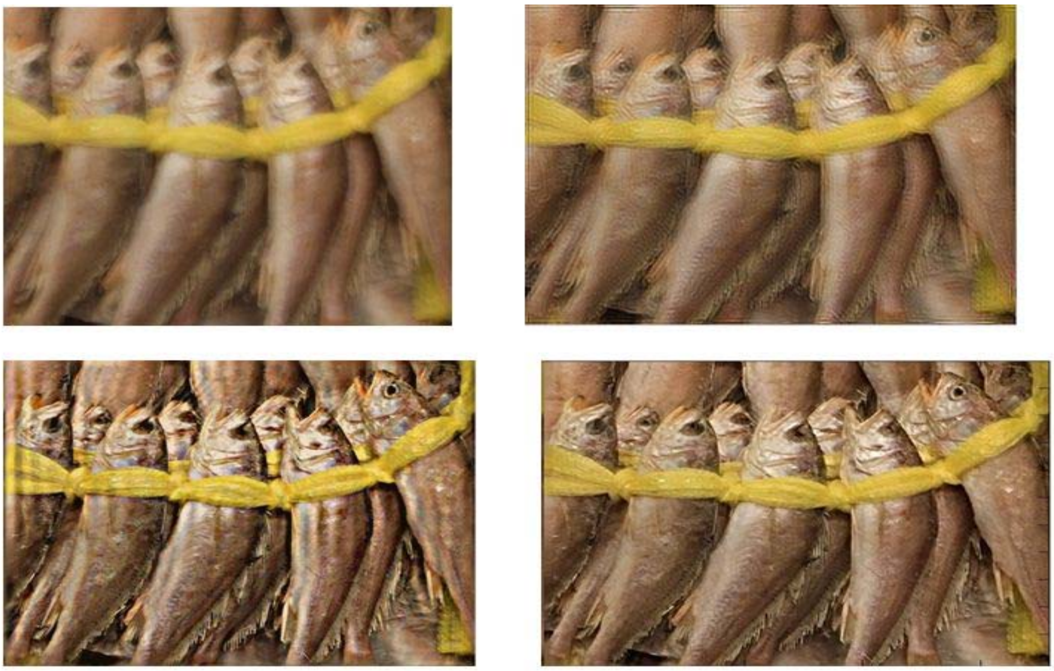

Fig.3. Deblurred Result: Top left: Input blurry image; Top right: Deblurred with algorithm of [3]; Bottom Left:Deblurred with algorithm of [9]; Bottom right: Deblurred with our Algorithm

We compared our algorithm with result of Fergus.et.al [3] and Filip.et.al [9] using the code provided online. The result of [3] and [9] shows artifacts. By examining the figures it can be seen that our deblurring results are better and visually pleasing than the previous works. Also our algorithm is faster than the previous algorithms such as [3]. Our algorithm takes only 3 minutes for deblurring an image of size 255 × 255 whereas algorithm of [3] takes 6 minutes and [9] takes 5 minutes.

V. CONCLUSION

This paper presents a novel algorithm to deblur image and video sequences. The main contribution of our work is the scale invariant regularizer that stabilizes the kernel estimation process. This regularizer is non convex so a minimization scheme that amounts to solving a series of l_1 problems with different regularization parameters is also introduced. From the simulation results it can be seen that the resulting algorithm is fast and robust.

Acknowledgment

We would like to express our sincere thanks to Mr.Geevarghese Titus (Asst.Prof, Ajce) and Mr. Binu Mathew (Asst. Prof, Ajce) for their valuable help and support.

References

1. 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..
2. 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
3. S. Zhang, S. Cho and S. Lee, “Fast motion deblurring,” *ACM Trans. Graph.* vol. 28, no. 5, p. 145, Dec. 2009.
4. M. Ben-Ezra and S. K. Nayar, “Motion-based motion deblurring,” *IEEETrans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 689–698, Jun.2004
5. D.B.Lee, S.C.Jeong, B.C.Song,Y.G.Lee, “Video deblurring algorithm using accurate blur kernel estimation and residual deconvolution based on a blurred unblurred frame pair”. IEEE transactions on image processing, vol. 22, no. 3, march 2013
6. D.Krishnan and Rob Fergus “Blind image deconvolution using a normalized sparsity measure”, CVPR 2011.
7. Rony Ferzli and Lina J. Karam , “A no-reference objective sharpness metric based on the notion of just noticeable blur”, IEEE Trans. On image processing Vol.8, April 2009
8. Dilip Krishnan and Rob Fergus, “Fast image deconvolution using hyper-laplacian priors”, NIPS 2009
9. Jan Kotera, Filip Sroubek, Peyman Milanfar Blind deconvolution using alternating maximum a posteriori estimation with heavy-tailed priors", CAIP 2013