International Journal of Computer Applications (0975 – 8887)
Volume 130 – No.13, November 2015

An Adaptive and High Quality Blind Image Deblurring using Spectral Properties

Seethu George
PG Scholar
Dept. of CSE,
VJCET, Vazhakulam, Kerala, India.

Resmi Cherian
Asst. Professor
Dept. of CSE,
VJCET, Vazhakulam, Kerala, India.

ABSTRACT
Blurring is a common artifact that produces distorted images with unavoidable information loss. The Blind image deconvolution is to recover the sharp estimate of a given blurry image when the blur kernel is unknown. Despite the availability of deconvolution methods, it is still uncertain how to regularize the blur kernel in an effectual fashion which could substantially improve the results even when the image is blurred to its extend. This paper presents a novel deconvolution method that describes an efficient optimization scheme that alternates between estimation of blur kernel and restoration of sharp image until convergence. The system engenders a more efficient regularizer for the blur kernel that can generally and considerably benefit the solution for the problem of blind deconvolution. Also the blur metric concept in the system provides an automated environment for the selection of deconvolutoin parameters. The outlier handling model used in this work detects and eliminates the major causes of visual artifacts. As a result the system produces high quality deblurred results that preserves fine edge details of an image and complex image structures, while avoiding visual artifacts. The experiments on realistic images show that the proposed deconvolution method can produce high quality deblurred images with very little ringing artifacts even when the image is severely blurred, and the ability of system in choosing the appropriate input parameters for deconvolution.

General Terms
Image processing, signal processing, blind and non blind image debluring

Keywords
Image deblurring, blind deconvolution, blur kernel estimation, point spread function, spectral methods, outlier detection, blur metric.

1. INTRODUCTION
One of the most common problem found in the area of digital image photography is image deblurring, which is to recover the sharp version of a given blurred or distorted image. The three main causes of blurry images are, out of focus, object motion, or by the camera moves while the shutter is open. In many situations recovering of the sharp version of the blurred image is necessary so that the details become recognizable to human eyes.

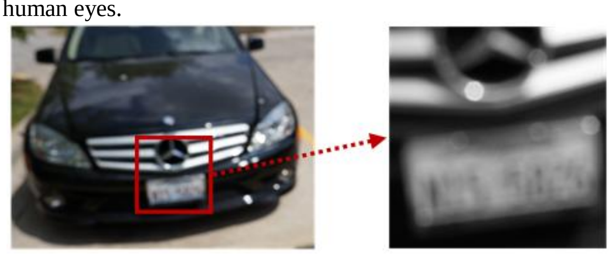


Fig 1: An image deblurring scenario. The picture is taken by using a high dimensional camera. The image got blurry due to a wild camera motion. This image needs to get processed so that human eyes are able to recognize the details in it (e.g., the number plate digits).

Recovering a sharp image from a single, motion-blurred or out of focused photograph has long been a fundamental and essential research topic in digital imaging. As the problem of deconvolution is ill-conditioned, an effectual criterion concerning to both the sharp image and blur kernel estimation are required to confine the space of candidate solutions. Blur kernel plays the key role of any deconvolution task, which will act as a mask during deblurring process to deconvolute the blurry image for obtaining its sharp version. So the main focus of blind deconvolution deals with estimating an accurate blur kernel for the given blurry image and then performing an appropriate deconvolution method.

Image deconvolution methods can be further divided into two major classes: **Nonblind**: which assumes that the blurring operator to be known. **Blind**: this assumes that the blurring operator is unknown. The foundations are based upon the suite of methods that are designed to remove or reverse the blurring present in the digital images. Various systems that can handle blurring caused due to different circumstances such as camera motion, object motion, etc are currently available. But one of the central challenge in image deblurring is to develop a method that can disambiguate between multiple solutions and then bias the deblurring process towards more likely result provided some prior information. Recent techniques adopt an additional regularizer for the blur kernel to accurately estimate the kernel for better quality output. However severely blurred images still pose a big challenge to the researcher community.

The blind deconvolution problem has been being widely investigated for several decades in signal processing, computer vision, computer graphics and image processing. A straightforward approach for blind deconvolution is to jointly seek for the sharp version of I_0 and a corresponding blur kernel, denoted as $K_0 \in R^{m1 \times m2}$:

$$B \approx I_0 * K_0 \qquad s.t \quad K_0 \in S \tag{1}$$

where * denotes the discrete 2D linear convolution operator, S is the simplex (non-negative and sums to one) of all possible blur kernels, where m_1, m_2 denotes the size of kernel and n_1, n_2 are the image sizes. In this way, the blind image deblurring problem is mathematically formulated as a blind deconvolution problem of recovering I_0, the latent sharp image when the generic blur kernel K_0 is unknown. It is in general difficult to correctly estimate the accurate kernel if both motion and the scene geometry are entirely unknown.

2. RELATED WORK

Early approaches usually assign simple parametric models for the kernel like a low pass filter in the frequency domain or a sum of normal distributions. Q. Shan et.al [5] has proven that it is possible to achieve a high quality deconvolution results in low computation time using some efficient optimization based deblurring method. This work accelerates latent image estimation in the iterative deblurring process without degrading the accuracy of kernel estimation, by combining a prediction step with simple non-blind deconvolution. Fast Motion Deblurring [6] is another method which produces deblurred result within a few seconds for a moderate sized image using some iterative deblurring process. The method consists of simpler steps and the system can recover the sharp version of a blurred image very fastly than any other method. Jinshan Pan and Zhixun Su [9] suggested a method, Fast ℓ_0-regularized kernel estimation which estimates a blur kernel from a single blurred image by regularizing the sparsity property of natural images. This method is able to restore useful salient edges for kernel estimation. Ringing is one of the most disturbing artifacts in the image deconvolution. By using a progressive inter-scale and intra-scale deconvolution [7], it is possible to recover visually pleasant images with very little ringing. The main advantage of this method is that it preserves the edges and reduces the ringing artifacts, especially for the large kernel.

The direct approach for problem of blind deconvolution is to jointly seek the sharp version of I_0 and the blur kernel K_0 by a minimizing function,

$$\min_{I,K} \|B - I * K\|_F^2 + \lambda f(I) \qquad (2)$$

where $\lambda \geq 0$ is a parameter and $\|.\|_F$ denotes Frobenius norm of a matrix. Usually the regularizer *f(I)* is chosen as the total variation. However, such image gradient based regularizers are generally critical and only favor a blurry solution over a sharp one, i.e., it is necessary to consider a modified or an extended version of the regularizer like,

$$\min_{I,K} \|B - I * K\|_F^2 + \lambda f(I) + \alpha h(K), \qquad s.t \quad K \in S \qquad (3)$$

While it is well-understood that the regularizer h(K) is vital and very crucial for blind deconvolution, the existing suggestions for h(K), e.g., the Sparse regularizer, Bayesian prior and the Gaussian function, in fact hold no real information about the desired blur kernel K_0. Therefore, in order to generate a useful solution, the existing deblurring approaches often require some heuristic regularizers such as salient structure selection. However those heuristic regularizers could work well only for some simple cases where there is no severe blur, but could not handle the difficult deblurring situations where the images are seriously blurred.

3. PROPOSED SCHEME

This work derives a much more effective blind deconvolution method that can efficiently handle most difficult deblurring tasks such as complicated blur kernels, comparing with previous methods. Even when the inputted image is blurred to the extent where details are not recognizable by human eyes, it is still possible for the proposed system to restore the sharp image with recognizable details. The new deblurring can be effectual for various blur situations such as defocus and motion blur. Unlike previous regularizers which may contain no real information about the blur kernel, the newly proposed regularizer has a strong outcome and can even directly regain the blur kernel K_0 without knowing the sharp image I_0. The new regularizer is based on a well-known observation; that is, sharp images are often high-pass and blurry images are usually low-pass, or in other hand, blurring will mostly decrease the image frequencies in Fourier domain.

The current deconvolution methods can perform well only if both the blur kernel contains no error and the blurry image contains no noise, which will result in common artifacts found in current deblurring methods. Therefore the proposed system introduces a better model that explicitly handles visual artifacts caused by deconvolution and an advanced iterative optimization that alternates between the latent image restoration and blur kernel assessment until convergence. The system also presents an analysis of the major deconvolution parameters that explicitly control the blind deconvolution and blur estimation process. Based on which a blur metric concept is provided so that the parameter identification can be automated. The flow chart representing the working of the proposed system is given in Fig. 2

Stepwise description of proposed system is as follows:

- Input Blurry image.
- Estimates blur metric for the image to give amount of blurriness.
- Use blur estimate to identify the parameters.
- Generate a Hessian matrix with estimated convolution matrix, eigenvalues and eigenvectors.
- Define the regularizer for the blur kernel using the Hessian matrix H.
- With such a strong trained regularizer $h^{L(B)}(K)$, together seek the sharp estimate I_0 and the blur kernel K_0 by resolving the optimization problem.
- Remove visual artifacts from the deconvolution result by making use of an efficient outlier handling method.

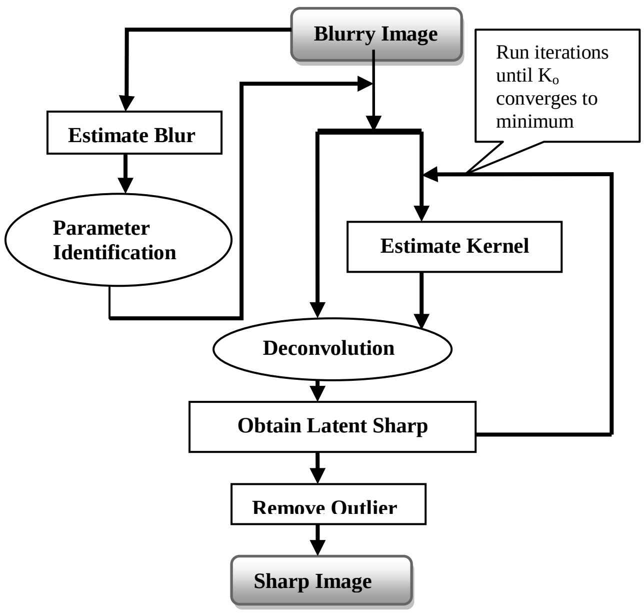


Fig 2: Flow chart of the proposed deblurring system

4. METHODOLOGY

The entire system working can simply modularized as follows based on the order of implementation.

4.1 Blur Estimation

There are four parameters in this blind deblurring algorithm: The kernel sizes m_1, m_2 ($m_1 = m_2$ usually), the sampling sizes s_1, s_2, and the tradeoff parameters α, λ. When the kernel sizes have been determined, the sampling sizes could be simply set as $s_1=1.5m_1$ and $s_2 = 1.5m_2$. The parameter λ plays the role of suppressing possible artifacts arising from the nonblind deconvolution procedure. Usually, it is moderately good to choose this parameter from the range of 0.001 to 0.002. The parameter α also needs to be set properly. When α is too small, the algorithm generally converges to the no-blur explanation. If α set to too large, the recovered image would be very sharp, but contains considerable artifacts (see Fig. 4).

In the existing system the parameters of the baselines are need to be manually tuned to best. But the proposed system is capable of obtaining all these parameters automatically by making use of the concept of Blur metric introduced by Frederique Crete-Roffet, Dolmiere T et al.[3], which will estimate the amount of blurriness of an image. The blur metric compute the blur annoyance of an image by reblurring it and comparing the distinctions between neighboring pixels just before and after the blurring. The description is summarized in Fig. 3.

The blur metric will be further used for the purpose of identification of the parameters for deconvolution. The algorithm for the Blur metric works as shown in the flow chart (Fig. 3).

Algorithm 1: Blur Metric Estimation

Let I denote the input image of size of m × n pixels. Steps are as follows;

1. The first step to estimate the blur annoyance of I is to re-blur it in order to obtain a blurred version B. A vertical and a horizontal strong low-pass filter have chosen to model the blur effect and to produce B_{Hor} and B_{Ver}.

$$H_v = 1/9 \times [111111111]; \qquad H_h = H_v'$$
$$B_{Ver} = H_v I; \qquad B_{Hor} = H_h I \tag{4}$$

2. Calculate the absolute difference images of blurred image B and input image I both horizontally and vertically, which is denoted as D_B_{Hor}, D_B_{Ver}, D_I_{Hor} and D_I_{Ver} in the image.
3. Evaluate the variations of the neighboring pixels i.e. $D\_I_{Ver} - D\_B_{Ver}$ and $D\_I_{Hor} - D\_B_{Hor}$. If this variation is slight, the initial image was already blurred whereas the initial image was sharp if the variation is high. This variation is calculated only on the absolute differences which have decreased.
4. In order to evaluate the variations from the initial image, calculate the sum of the coefficients of V_{Ver}, V_{Hor}, D_I_{Ver}, D_I_{Hor} to obtain S_I_{Ver}, S_I_{Hor}, S_V_{Ver}, S_V_{Hor}.
5. Normalize the result in a desired range from 0 to 1 by,

$$b\_I_{Ver} = \frac{s\_I_{Ver} - s\_V_{Ver}}{s\_I_{Ver}}, \qquad b\_I_{Hor} = \frac{s\_I_{Hor} - s\_V_{Hor}}{s\_I_{Hor}} \tag{5}$$

6. Select the more annoying blur among the horizontal one and the vertical one as the final blur value,

$$blur\_I = Max\,(b\_I_{Ver}\,,\, b\_I_{Hor}) \tag{6}$$

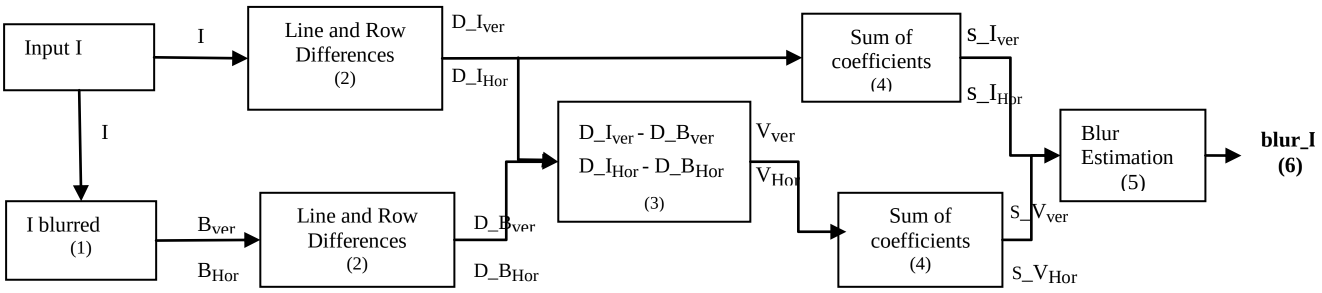


Fig 3: Flow chart of the blur metric algorithm with the equations references

4.2 Parameter Identification

After the blur metric estimation, the next step is to select the deconvolution coefficients α, λ, kernel size and sample rate. Based on the observations on the various tests that have been conducted the parameter selection table is designed as shown below.

Table 1. Deconvolution parameter selection table

Blur Value	Alpha	Kernel Size	Lambda	Sample Rate
Above 0.7	4000	[17 17]	0.001	0.75
0.6 - 0.7	90	[13 13]	0.001	0.75
0.5 - 0.6	70	[13 13]	0.001	0.75
0.4 - 0.5	1200	[13 13]	0.001	0.75
0.3 - 0.4	900	[13 13]	0.001	0.75
Below 0.3	400	[13 13]	0.001	0.75

The blur value less than 0.3 cannot be considered as really blurred image as there exist some images which are originally smooth itself. The blur value range is always found to be greater than 0.3 for really blurred image and the blur value greater than 0.7 is considered to be severely blurred case. The α value actually plays the role of controlling the sharpness of an image. For each blur value that is calculated, α need to be set properly as shown in the above table. The kernel size should be large for severely blurred images i.e. kernel size is set to 17×17 for a blur value greater than 0.7 and for all other values a 13×13 kernel is used to aid the parameter selection process more convenient based on experiments conducted. The λ value can vary between 0.001 to 0.002. For a range 0.001 to 0.0015 of λ the output is same. So the λ value is set to 0.001 for all images, which can produce a quality output. The sample size is set to $1.5m_1$ for all cases, where m_1 is the size of the kernel ($m_1 \times m_2$), assumes $m_1=m_2$.

4.3 Hessian Matrix Creation

As stated by Guangcan Liu, Shiyu Chang and Yi Ma in[1], the classical observation suggests that the spectrum (the Fourier frequencies, i.e. the set of eigenvalues) of the linear operator for a blurred image should be considerably smaller than that for its sharp version. Based on this observation, a typical convex kernel regularizer which tends to be minimal at the desired blur kernel K_0 can be devised. The regularizer denoted as $h^{L(B)}(K)$ is designed by using the extracted features of the given blurry image. The Hessian matrix is the means by which the extracted features will be used in the regularizer.

4.3.1 Feature Extraction

The convolution operator is linear, so it can be converted into matrix multiplication form. Let *v*(.) denotes the vectorization of a matrix, then it can be rewritten like,

$$v(X * Y) = A_{k1k2}(X)\, v(Y) \tag{7}$$

where A_{k1k2} (.) is called the convolution matrix of a matrix, and the suffix k_1, k_2 are taken as parameters. For an ℓ_1-by-ℓ_2 matrix X, its convolution matrix, denoted as $A_{k1k2}(X)$, which is of size $(l_1+k_1-1)(l_2 + k_2 -1)$ - by - k_1k_2.

In this work, an image *I* is considered as a matrix associated with some feature filter L : $L(I) = L * I$. The choice for the feature filter is $L = LoG$, which extracts the edge features of an image i.e. this work use edge features by default. The spectrum of an image in the edge domain is more susceptible to blurring than in the raw pixel domain, and thus the blur kernel K_0 is easier to restore by using edge features than using raw pixel values.

So called spectral properties such as convolution eigenvalues and eigenvectors of an image can be represented by a feature filter *L*, denoted as $\{\sigma_i^L(B)\}$ for eigenvalues and $\{k_i^L(B)\}$ for eigenvectors. The convolution eigenvectors/ eigenvalues are exactly the right singular vectors/values of the convolution matrix. So, for an image I with the associated convolution matrix $A_{s1s2}(L(I))$, its all s_1s_2 convolution eigenvalues and eigenvectors can be found by computing the Singular Value Decomposition (SVD) of the matrix, $(A_{s1s2}(L(I)))^T A_{s1s2}(L(I))$, which is of size s_1s_2-by-s_1s_2. Hessian matrix H is generated by making use of these convolution eigenvalues and eigenvectors. So the regularizer for the kernel can be defined using H as: $h^{L(B)}(K) = (v(K))^T H v(K)$ with the Hessian matrix H given by,

$$H = \sum_{i=1}^{s1s2} \frac{(A_{m1m2}(k_i^L(B)))^T A_{m1m2}(k_i^L(B))}{(\sigma_i^L(B))^2}, \tag{8}$$

where $A_{m1,m2}(k_i^L(B))$ is the convolution matrix of the i^th^ convolution eigenvector of B, $\sigma_i^L(B)$ is the convolution eigenvalue of B, $\{m_1,m_2\}$ are the sizes of the blur kernel. Algorithm 2 summarizes the whole procedure of computing the Hessian matrix H.

Algorithm 2 : Computing the Hessian Matrix

Input : Blurry image B.
Parameters : Kernel size(k_1, k_2); Sampling size(s_1,s_2).

1. Compute the edge map of B by L(B) = B * LoG
2. Compute the convolution matrix $A_{s1s2}(L(B))$
3. Let $M = (A_{s1s2}(L(I)))^T A_{s1s2}(L(I))$. Then obtain convolution Eigenvalues $\{\sigma_i^L(B)\}$ and Eigenvectors $\{k_i^L(B)\}$ by performing SVD on M.
4. For each $\{k_i^L(B)\}$ compute its convolution matrix $A_{m1,m2}(k_i^L(B))$
5. Compute the Hessian Matrix H

Output : H

4.4 Kernel Estimation and Fast Deconvolution

The blind deconvolution problem can be successfully handled by an effective regularizer that is defined for the blur kernel in this work. The inclusion of the classical observation that the spectrum or the set of eigenvalues of the linear operator of a blurry image should be significantly smaller than that for its sharp counterpart hold the main part of the entire process. Based on this observation, a regularizer is designed which tends to be minimal at the desired blur kernel K_0. That is, given an observed image B characterized by a certain image feature L, then the convex regularizer function can be written as $h^{L(B)}(K)$.

Unlike most available methods where the regularizer h(K) is independent of the observed blurry image B, this regularizer $h^{L(B)}(K)$, which explicitly depends on the given blurry image and also provides strong information about how the blurry image is related to the sharp image I_0. So the desired blur kernel K_0 can be exactly retrieved by directly minimizing $h^{L(B)}(K)$. Equipped with such a powerful regularizer $h^{L(B)}(K)$, it can jointly seek the blur kernel K_0 and the sharp image I_0 by solving the optimization problem.

Unlike the previous algorithms which need to carefully choose the initial solution, this system simply choose the observed blurry image B as the initial condition for *I*. Then blind deconvolution is carried out by iterating the following two procedures until convergence:

1. While fixing the variable *I* (the latent sharp image), the blur kernel K updated by solving $\min_K \| B - I * K \|_2^F + h^{L(B)}(K)$, as stated by Guangcan Liu et.al in [1] which is equal to the following quadratical program:

$$\min_K \left\| v(B) - A_{m1,m2}(I) v(K) \right\|_2^2 + \alpha (v(K))^T H v(K) \tag{9}$$

   where the hessian matrix H is calculated by Algorithm 2, $\|.\|_2$ is the $\ell 2$-norm of a vector. The minimization is done by using an optimization tool quadprog of Matlab.

2. While fixing the blur kernel K, the sharp version of the blurry image *I* is updated by,

$$\min_I \left\| B - I * K \right\|_F^2 + \lambda \left\| \nabla I \right\|_1 \tag{10}$$

   which can be solved by any of the existing non-blind deconvolution algorithms. This work simply use the fast deconvolution method introduced by Krishnan and Fergus [2].

The nonblind deconvolution method using Hyper-Laplacian Priors can produce output in ~3 seconds. According to Krishnan and Fergus, if x is the original uncorrupted grayscale image of N pixels; y is an image degraded by blur, which is assumed to be produced by convolving x with a blur kernel k and adding zero mean Gaussian noise, then the deconvolution can be done by using the minimization scheme,

$$\min_x \sum_{i=1}^{N}\left(\frac{\lambda}{2}(x\oplus k - y)_i^2 + \sum_{j-1}^{J}\left|(x\oplus f_j)_i\right|^{\alpha}\right) \qquad (11)$$

where i is the pixel index, and $\oplus$ is the 2-dimensional convolution operator. For simplicity, two first-order derivative filters are used, f_1 = [1 -1] and $f_2 = [1\ \ -1]^T$.

In addition, unlike previous blind deblurring methods that needs to carefully control the number of iterations, the proposed system run the iterations until convergence. The optimization scheme for deconvolution alternates between estimation of blur kernel and sharp image restoration until convergence. Usually, the algorithm converges within the maximum limit of 200 iterations.

4.5 Outlier Removal

The existing system optimization approaches with image priors often produce severe ringing artifacts in the deconvolution result even when the blur kernel is already known or well estimated. This is mainly because the blur model does not consider nonlinear outliers that often present in real imaging process. Fig. 4 shows the result of the existing system that contains severe visual artifacts which appears as horizontal and vertical lines in the image. Removing these outliers are extremely hard using image processing techniques.

An efficient algorithm introduced by S. Cho, J. Wang, and S. Lee [4] that handles outliers explicitly in the deconvolution process is used here. For this purpose the algorithm categorize image pixels into two major categories: **inlier** pixels that satisfy the linear blur model and can be well recuperated utilizing traditional deconvolution methods, and the **outlier** pixels which cannot be elucidated by the linear model. For classification, a binary map m is used, such that $m_x = 1$ if the observed intensity b_x is an inlier otherwise $m_x = 0$, where the subscript x denotes the pixel index. Given the blurred image b containing outliers and the blur kernel k, exclude the outliers from the deconvolution process by making use of the inlier map m, to find the most accurate latent image *l*. Since the true value of m is unknown, an expectation maximization (EM) method which computes the expectation of m alternately and performs deconvolution using the expectation is used here.

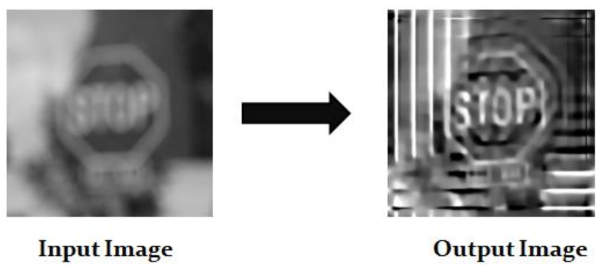


Fig 4: The existing system output containing outliers

For inliers ($m_x = 1$), b_x has a Gaussian noise,

$$p(b_x \mid m_x=1, k, l) = N(b_x \mid m_x=1, (k * l)_x, \sigma) \qquad (12)$$

For outliers ($m_x = 0$), bx may have an arbitrary value within the dynamic range (DR = [0 1], here).

$$p(b_x \mid m_x=0, k, l) = 1/\mid \text{DynamicRange} \mid \qquad (13)$$

Consider $p(m_x \mid f_x)$, the probability based on the value of f_x as.

$$p(m_x = 1 \mid f_x) = \begin{cases} P_{in} & \text{if} \quad f_x \in DR, \\ 0 & \text{otherwise} \end{cases} \qquad (14)$$

where DR is the dynamic range, and $P_{in} \in [0, 1]$ is the probability that b_x is an inlier. For example, by setting P_{in}=0.9, assume that 90% of non-clipped observed pixels b_x are inliers. According to the blur model, when f_x is out of DR, the observed intensity b_x cannot be an inlier, which is either an outlier of another type or a clipped value, thus always m_x should be 0 in this case. The following derives the details of the two steps of the algorithm.

E step

Using the current estimate l_0 of l and by taking N as a Gaussian distribution, and σ the standard deviation as 5/255, calculate

$$E[m_x] = \begin{cases} \dfrac{\mathcal{N}(b_x \mid f_x^0, \sigma) P_{in}}{\mathcal{N}(b_x \mid f_x^0, \sigma) P_{in} + C P_{out}} & \text{if} \quad f_x^0 \in DR, \\ 0 & \text{otherwise} \end{cases} \qquad (15)$$

where $f_0 = k \star l_0$ and $P_{out} = 1 - P_{in}$. Actually the only value that is computed in E step is $E[m_x]$. This step will find the outliers present in the image by using the binary map m.

M step

The M step finds the amended estimate of l_n by updating the weights using the minimization function,

$$\sum_x w_x^m \left|b_x - k * l)_x\right|^2 + \lambda\psi(l) \qquad (16)$$

where,

$$\lambda\psi(l) = \sum_x \{w_x^h \left|(\nabla^h l)_x\right|^2 + w_x^v \left|(\nabla^v l)_x\right|^2\} \qquad (17)$$

where $w_x^m = E[m_x]/2\sigma^2$, $w_x^h = \left|(\nabla^h l)\right|^{\alpha-2}$, $w_x^v = \left|(\nabla^v l)\right|^{\alpha-2}$

For fixed w^h and w^v, with respect to l , Eq. (16) becomes a quadratic function, which can be minimized effectively using the conjugate gradient method.

Algorithm 3 : EM Deconvolution for Handling Outliers

procedure DECONVOLUTION(b, k)

1. Let w_x^m , w_x^h & w_x^v←1 for all x
2. Set l_o by minimizing (20)
3. for iter = 1;N_iters do
4. E step updates w^m, w^h and w^v using l^o
5. M step updates l^n by minimizing (16)
6. $l^o = l^n$
7. end for
8. return l^o

end procedure

5. EXPERIMENTAL RESULTS

The experiments conducted on five examples are shown here(two synthetical, three real): The synthetical images are the convolution of 300×300 natural images and 13×13 synthetical blur kernels. The input blurry images (R1, R2, R4, R5) are published by Guangcan Liu et al.[1]. Figure 5 shows the comparison results. On the simple examples with easy blur kernels (last three examples in Fig. 5), the proposed algorithm performs really well as well as the existing systems. While dealing with the challenging examples (first two examples in Fig. 5), where the blur kernels are complicated, it can be seen that the proposed algorithm works distinctly better than the most competitive existing systems. In particular, the first and second examples illustrate that it is even possible for the proposed algorithm to successfully handle some extremely difficult cases, where the blurry images are very unclear such that human eyes are unable to recognize their contents. The ringing appeared on strong edges and textures are considerably reduced. The remaining artifact is mainly caused because the motion blur is not spatially invariant.

6. PERFORMANCE EVALUATION

To evaluate the deconvolution performance of the proposed system, SSIM based comparison with the existing system is used here. Structural SIMilarity index (SSIM) is used for measuring image quality by providing an image 'A' whose quality is to be measured with a Reference image against which quality is measured. Both images must be of the same size and class. The idea behind Structural similarity is that the high adaption of human visual system to the structural information of an image and the function attempts to measures the change in this information between the reference image and distorted image. Based on various tests, SSIM does a much better job at measuring the subjective image quality than MSE or PSNR.

Structural similarity index can be measured using a direct function called ssim() available in Matlab 2014 and above versions. Function call looks like,

[ssimval, ssimmap] = ssim(A, ref)

where ssimval is the global SSIM index value of the image and the ssimmap is displayed by calculating the local SSIM values for each pixel of the image A using ref as the reference image.

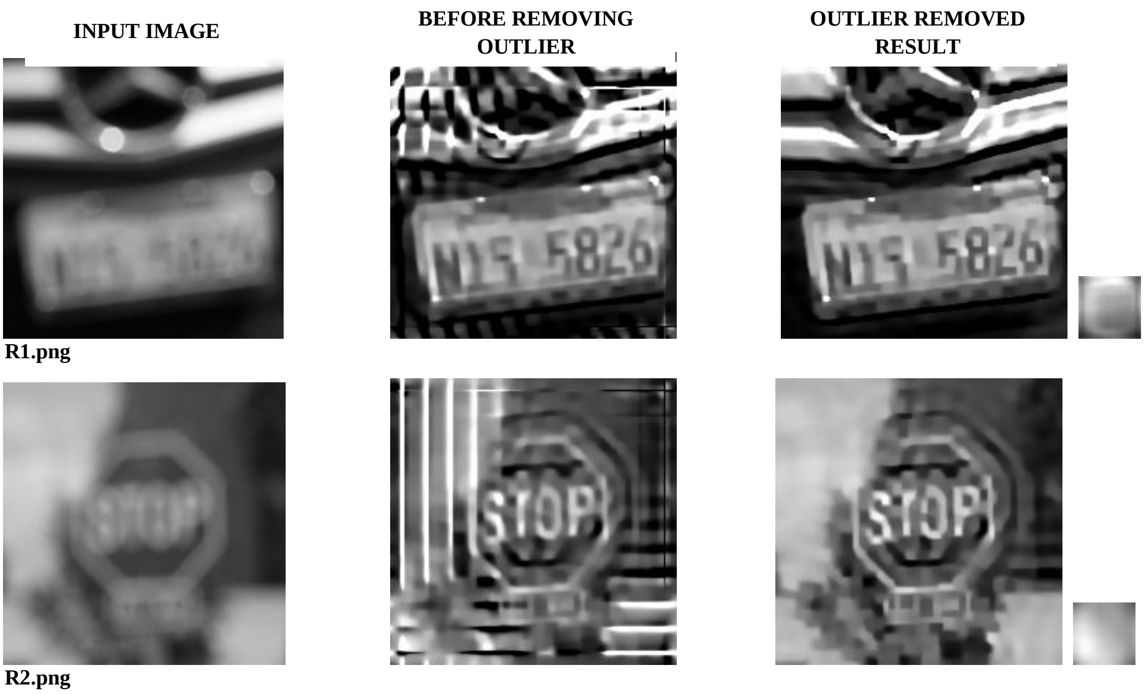
Higher the SSIM value higher will be the image quality. So the proposed system is evaluated by taking the SSIM values between the input image and the existing system result as well as between the input and the proposed system result. The calculations made here shows that the SSIM value between input image and proposed system output goes higher than that between input image and existing system output. This is because of the presence of outliers such as ringing artifacts and saturated pixels in the existing system result, which make the output image structure to be entirely different than that of the input image. Since the proposed system removes all such outliers explicitly it looks more similar to input image in structure, thus the SSIM value will be higher.

Table 2. Structure similarity comparison of input image with results

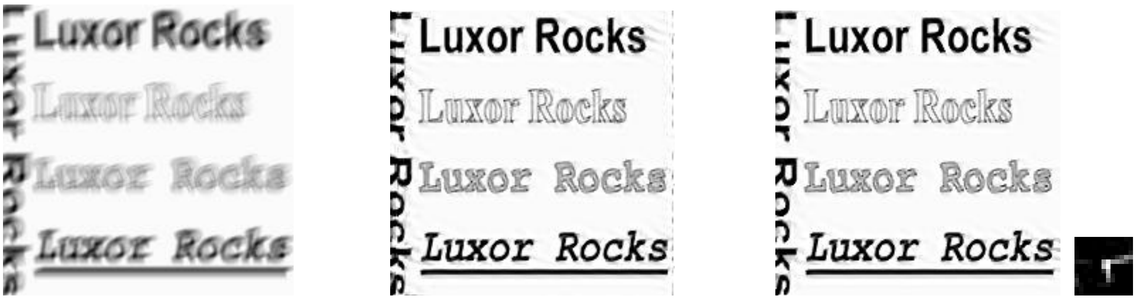
Image Name	SSIM value of image containing outlier with input	SSIM value of image without outlier with input
R1.png	0.2970	0.3909
R2.png	0.3022	0.6095
R3.png	0.6694	0.7531
R4.png	0.6973	0.7115
R5.png	0.5161	0.5235

The SSIM values of the proposed system and existing system are shown in the Table 2 (input Vs existing system SSIM value is on 2nd column and input Vs proposed system SSIM value is on 3rd column). After analyzing the results which clearly shows that the proposed system provides a very good result in terms of both sharpness and quality than the existing system.

The comparison with some previous methods(in Fig. 6) also shows that the capability of the proposed system to estimate an approximate kernel that can give a sharp outcome even when the image is severely blurred, mean while the previous methods could not. The Table 3 shows how effective the proposed system in reducing the amount of blurriness of the input image after deconvolution. The blur metric is used here



R3.png



R4.png

R5.png

Fig. 5. Comparison results. Left: The input blurry images (published in Guangcan Liu et al.[1] & Q. Shan et al.[5]). Middle: The existing system result containing outliers. Right: Proposed system result by removing outliers. Corresponding kernels are on the right most side.

as the means for measuring the blur annoyance of an image. The proposed method significantly reduces the amount of blurriness to 50% lesser for output image.

Fig 6: An output comparison with previous method (High Quality motion Debluring [5]) for severely blurred cases.

Table 3: Change in blur estimate before and after deconvolution

Image Name	Input Image Blur Estimate	Output Image Blur Estimate
R1.png	0.7773	0.4865
R2.png	0.7367	0.5312
R3.png	0.6092	0.3675
R4.png	0.3374	0.1479
R5.png	0.4592	0.2288

7. CONCLUSION

Recovering sharp version of a blurry image is a long standing problem for many image processing applications. Although there exist various systems that can effectively handle the blind deconvolution problem, they fail to handle the severely blurred cases as well as irrelevant visual artifacts and noises occurring in the deconvolution result. By defining an effective regularizer, the proposed system can appreciably benefit the solution of the blind deconvolution problem along with approximate kernel estimation. The blur metric that quantify the blur annoyance on a blurry image make the system capable of identifying the appropriative parameters which can bias the deconvolution process in the right way. The proposed system also ensures the quality of the output by detecting and explicitly handling outliers in the deconvolution process. Thus for the blind deconvolution problem the system is capable of dealing with the challenging examples where the blur kernels are complicated and perform really well than the existing methods in terms of both quality and sharpness of an image.

8. ACKNOWLEDGMENTS

First and Foremost the authors thank God Almighty for His divine grace and blessings in making all this possible. The authors gratefully acknowledge all the staff members of Department of Computer Science and Engineering, Viswajyothi College of Engineering and Technology, for providing their motivation, constant support and encouragement and for guiding with patience and enthusiasm. Also the authors thank students of M.Tech, Department of Computer Science and Engineering, Viswajyothi College of Engineering and Technology, Vazhakulam, for their assistance and help during the course of this project.

9. REFERENCES

[1] Guangcan Liu, Member, IEEE, Shiyu Chang, and, Fellow, IEEE, "Blind Image Deblurring Using Spectral Properties of Convolution Operators," *IEEE Transactions on image processing*, Vol. 23, NO. 12, December 2014.

[2] D. Krishnan and R. Fergus, "Fast image deconvolution using hyper-Laplacian priors," *in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2009, pp. 10331041.

[3] Frederique Crete-Roffet, Thierry Dolmiere, Patricia Ladret, Marina Nicolas."The Blur Effect: Perception and Estimation with a New No-Reference Perceptual Blur Metric," *SPIE Electronic Imaging Symposium Conf Human Vision and Electronic Imaging*, - GRENOBLE – 2007

[4] S. Cho, J. Wang, and S. Lee, "Handling outliers in non-blind image deconvolution," *in Proc. Int. Conf. Comput. Vis.*, 2011, pp. 495502.

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

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

[7] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum, "Progressive inter-scale and intra-scale non-blind image deconvolution," *ACM Trans. Graph*., vol. 27, no. 3, pp. 110, 2008.

[8] Mariana S., C. Almeida and Lus B. Almeida, "Blind and Semi-Blind Deblurring of Natural Images," *IEEE Transactions on image processing*, Vol. 19, NO. 1, January 2010.

[9] Jinshan Pan and Zhixun Su, "Fast ℓ_0 Regularized Kernel Estimation for Robust Motion Deblurring," *IEEE signal processing*, Vol. 20, NO. 9, September 2013.

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

[11] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman,"Removing camera shake from a single photograph," *ACM Trans.Graph.*, vol. 25, no. 3, pp. 787–794, 2006.