Image Deblurring with Compressive Sensing

Rahel Berhanu Dibeya[1,*], Fitsum Assamnew Andargie[1,*]

[1]*School of Electrical and Computer Engineering, College of Technology and Built Environment*
Addis Ababa University, Addis Ababa, Ethiopia
**Corresponding Author's E-mail address: mulrahel@yahoo.com, fitsum.assamnew@aait.edu.et*

DOI: https://doi.org/10.63990/zede.v43i.12974

ABSTRACT

Compressive sensing is a technique that enables recovery of signals represented by an underdetermined system of equations. Such a recovery of an original signal is made possible if the samples are represented in a sparse manner provided an appropriate measuring matrix is used for the modelled system. Blurred images are examples of signals that are sparse especially in transform domains. Different researches have been done to show the possibility of recovering blurred images that use sparse representation of transform domains by applying compressive sensing. In our work, however, we propose a model that doesn't require transforming into other domains. In addition, a box-wise approach has been used that derives the underdetermined system matrix from 7x7 segmented boxes of the blurred image. Compressive sensing algorithms are applied on these boxes to recover the whole image iteratively. Our method is shown to have a much better computational complexity than the traditional Lucy-Richardson deblurring method. Thus, with this improved computational complexity, the study provides an initial platform to deblur images using box-wise method and compressive sensing technique.

Keywords: Compressive Sensing, Deblurring, Gaussian blur, IHT, Sparsity.

1. INTRODUCTION

Image processing is a field which finds various applications in everyday life. In digital image processing, different operations are done one of which being image restoration. Image restoration is the process of recovering an original image from a noisy or blurred one [1]. One part of image restoration is image deblurring in which the blur is removed from a corrupted image by different techniques. In image deblurring, it is desired to have a restored image that is as much close to the original one by using minimal computational resources.

Presently, there exist different types of deblurring techniques. The Lucy-Richardson algorithm, neural network approach, deblurring with Noisy Image pairs and deblurring with Handling Outliers are few of them [2]. Among the different types of blurs encountered in image processing, Gaussian blur is one of them. For instance, Gaussian blur occurs in images taken in astronomy or medicine such as MRI.

Compressive sensing is a method which allows finding solution of equations which are underdetermined [3]. For reconstruction to be effective in compressive sensing, sparsity of the underdetermined system is one of the preconditions [4]. Blurred images are usually sparse in transform domains [5]. In this paper, a method which removes Gaussian blur using compressive sensing is introduced. With this method, a sparse system is derived and solved without transforming into other domains.

This research paper contains five parts. The first part is the introduction. In the second part, an overview of compressive sensing including definition, existing algorithms and related works are discussed. The third part details the modeling process used in this work. The fourth part includes the results obtained by applying the compressive sensing algorithm and their comparison with other deblurring methods. Finally, conclusion and recommendation are given in the fifth part.

1.1 Compressive Sensing Overview

Compressive sensing is a theory first developed by David L. Donoho [3] and Candes, Romberg and Tao [6] in 2006 that can be used to solve a system of equations which is underdetermined. Starting from its first development, various algorithms have emerged that enable to solve an underdetermined system of equations. These algorithms have preconditions that need to be met for successful recovery of the solution. The first precondition is that the underdetermined system of equations should be represented in sparse manner [7]. The other common precondition for faithful recovery is incoherence which requires the matrix that represents the underdetermined system of equations to have low coherence [8]. The coherence of a matrix is defined as the absolute value of the maximum cross-correlations between its columns [9].

Let the underdetermined system of equations be represented by $\phi(mxn)$ which is obtained by multiplying some original set of equations, $\psi(nxn)$, by a matrix $M(mxn)$ known as a measuring matrix where $m<<n$. And let the data to be recovered be a vector X of length n. Then the system of equations which is underdetermined can be represented as [3]:

$$Y = M\psi X \,, \quad (1)$$

where ψ: *nxn* matrix

M: *mxn* matrix

X: n sized vector to be restored

Y: m sized known vector a.k.a observation vector

$$Y = \phi X \quad (2)$$

where $\phi = M\psi$: *mxn* matrix

For successful reconstruction to take place by compressive sensing, the matrix ϕ which is the product ψ and M should be sparse and have low coherence [7][8].

1.2 Restricted Isometry Property (RIP)

In compressive sensing, a parameter known as Restricted Isometry Property (RIP) must be satisfied to ensure the faithful recovery of signals. The importance of this parameter relies on the fact that low coherence of the ϕ matrix is closely related to it [8]. The Restricted Isometry Property of the matrix, ϕ, given by $\phi = M\psi$, is defined as follows:

Assume a matrix $\phi_{mxn,}$ and an integer s such that $1 \le s \le n$. If a constant $\delta_s \in (0,1)$ exists such that, for every $m\ x\ s$ submatrix ϕ_s of ϕ, and for every vector z of dimension s [4],

$$(1-\delta_s)\|z\|_2^2 \le \|\phi_s z\|_2^2 \le (1+\delta_s)\|z\|_2^2 \quad (3)$$

holds true, then the matrix ϕ will have restricted isometry property of order s with the restricted isometric constant δ_s.

Most compressive sensing algorithms specify that the RIP value of the matrix ϕ to be less than a certain constant so that a successful reconstruction takes place. For example, Cande [8] specifies that the restricted isometric constant of order 2s should satisfy $\delta_{2s} < \sqrt{2} - 1$ for successful recovery in compressive sensing.

1.3 Sparsity

A vector or a matrix is said to be sparse if it consists of mainly zero elements. A vector or a matrix is said to be s-sparse if it has utmost s non-zero elements [10]. Sparsity enables to bring about efficient solutions in compressive sensing and different algorithms depend on it to recover a signal [7][11]. For a matrix $\phi_{mxn,}$in (2), Donoho and Tanner [12] state that compressive sensing algorithms can recover most sparse signals if s is given by:

$$s = \frac{m}{2\log n} \quad (4)$$

1.4 Incoherence

In compressive sensing, it is important that the measurement matrix M is selected in a way such that it has the lowest possible coherence with ψ [9]. The mutual coherence, μ, for the matrix $\phi = M\psi$ which needs to be minimized is described by the expression given below [8].

$$\mu(\phi) = \max_{i \neq j, 1 \leq i, j \leq n} \left\{ \frac{|\phi_i^T \phi_j|}{\|\phi_i\| \cdot \|\phi_j\|} \right\} \quad (5)$$

where ϕ_i is the i[th] column of ϕ.

In the work by Tropp [13][14], it is stated that the coherence value, μ, need to satisfy

$$s < \frac{1}{2}(\mu^{-1} + 1)$$
(6)

for an accurate reconstruction to take place in the compressive sensing algorithms known as Orthogonal Matching Pursuit and Basic Pursuit.

1.5 Reconstruction Algorithms

In compressive sensing, there are different categories of reconstruction algorithms. The Matching Pursuit and Iterative Thresholding algorithms are two of them. The Matching Pursuit class of algorithms tries to represent a signal by linear expansion functions that form a dictionary [15]. Then the Matching Pursuit algorithm optimally selects dictionary elements that can best approximate the signal. The Orthogonal Matching Pursuit (OMP) [16] and Compressive Sampling Matching Pursuit (CoSaMP) [17] are examples of Matching Pursuit algorithms.

The Iterative Thresholding algorithms try to recover a signal iteratively. They use a thresholding function $\mathbf{H}s(\mathbf{x})$ at each iteration to set components of a vector X which are less than some number ε to zero and leave the rest of the components untouched [18]. Iterative thresholding algorithms include the Iterative Hard thresholding algorithms and Iterative Soft thresholding algorithms.

1.6 Related Works

Image denoising based on compressive sensing was done by A. Tavakoli and A. Pourmohammad[19] in which an additive noise was used to model the compressive sensing equation by $Y = \phi(X + Z)$ .The authors performed compressive sensing denoising using existing algorithms namely, Orthogonal Matching Pursuit(OMP) and Iterative Hard Thresholding (IHT) in which they illustrated that IHT is faster than OMP. They also compared compressive sensing denoising with classical filters like Wiener filter, Median filter, Wavelet denoising and Gaussian filters. And the results showed that compressive sensing denoising gave the same result as some of the classical filters or fairly better result than the rest of the existing methods.

In the work by Bruno Amizic et al. [5], blind image deconvolution is performed using compressive sensing. The authors experimented to show that a blurred image is mostly compressible in the transform domains. Based on this fact, the authors proposed a new algorithm that solves a constrained optimization problem. In doing so, they extended compressive sensing algorithms for use in blind image deconvolution and the experimental results from the work shows fairly better outputs than that of existing algorithms such as CoSamp.

Blind image deblurring using compressive sensing has also been performed by J. Yu et al. [20]. The work exploited the fact that similar structures usually recur in a natural image. The authors also exploited the fact that a natural image exhibits multiple similar patches or structures when the image is down sampled. Thus, the authors used the down sampled version of the blurred image in order to find sparse representation of the original image. Using structural multi-similarity and sparse representation a blind motion deblurring method was developed which was shown to have 98.88% success rate.

Metzler et al. [21] applied compressive sensing to already existing denoising methods. The authors integrated the existing denoising framework AMP (Approximate message passing) with compressive sensing recovery. In this method, they illustrated that using Denoising AMP and compressive sensing together gives state of the art recovery while operating ten times faster than existing denoising algorithms.

Compressive sensing image denoising is also done by Kang et al. [22]. In this work, the image was decomposed into edge and flat regions. In addition, an 8x8 measurement matrix was designed which was applied to the first three wavelet coefficients of the blurred image. Then from the existing compressive sensing algorithms, OMP (Orthogonal Matching Pursuit) was applied to construct each block in the image. Different error thresholds were used based on the block being in edge or flat region. Based on the experiments done by the authors, the proposed method gives better results than other existing methods.

2. MATERIALS AND METHODS

2.1 Applying Compressive Sensing to Deblur Gaussian Blur

Compressive sensing allows to solve an underdetermined system of equations given the sparsity and incoherence conditions are satisfied. It is known that there are different existing methods for sparse representation of signals. One of these methods commonly used for sparse representation is transformation of the blurred image into domains such as wavelet [5]. In this work, a non-blind deblurring that doesn't require transforming the blurred image into another domain has been done. The deblurring method is a non-blind one with a known kernel. The chosen kernel is a two-dimensional 7x7 Gaussian kernel with standard deviation of $\sigma = 2$.

2.2 Representing Gaussian Blur by a Sparse Matrix Using Convolution

In deriving a model for the deblurring, image convolution with the above-mentioned kernel is utilized. Convolution of an image involves replacing the pixels in an image with the linear combination of the neighbouring pixels according to the values in a certain kernel [23].

By applying the standard procedure of convolution between the stated Gaussian kernel and an image with width w and height h pixels, there will be a resulting wxh system of equations. Thus, given an image blurred with this kernel, if the original image is required to be restored, the following system of equations needs to be solved.

$$Y = \psi X, \tag{7}$$

where Y - the blurred image vector of length $s=wxh=wh$

ψ - an $s \ x \ s$ matrix such that $s= wxh=wh$

X - the original image vector of length $s=wxh=wh$ to be restored

Basically, wxh is the size or the total number of pixels of the image. Therefore, there will be total number of equations which is equal to the total number of pixels in the image. To solve these equations in the direct way, the inverse of the matrix ψ has to be solved which usually has computational complexity of $O(s^3)$, where $s= wh$.

As mentioned earlier, the system of equations $Y = \psi X$ is obtained by convolving each pixel of the image with the convolution kernel. In these equations, a small fraction of the total number of pixels (7x7 or 49 neighboring pixels) is used to replace a pixel with their linear combinations. Because of this, the resulting equations are sparse mainly consisting of zeros. The sparsity found in these equations has made it possible to apply compressive sensing algorithms without transforming into other domains.

2.3 Specification of the Approach Used for Compressive Sensing Image Deblurring

While it is possible to apply compressive sensing algorithms to the whole set of equations which represents the total number of pixels, in this work however, an alternative method has been used. In this approach, the image has been segmented into 7x7 boxes measured in pixels. Then, 49 equations were derived resulting from convolution of every pixel in each box with the 7x7 Gaussian kernel. Out of the 49 equations, about half of them will be selected by a measurement matrix M. And to these selected equations, a compressive sensing algorithm is applied to retrieve the 49 original pixels. Finally, this procedure is repeated at each box iteratively until the whole image is covered.

The reason behind following this approach was initially to simplify the process of computing the RIP of the matrix to which compressive sensing was to be applied. In the end, however, the computation of the RIP parameter was not found necessary. This was because the Hard iterative Thresholding algorithm used in the work was found to converge because the norm-2 of the matrix involved was less than one.

Derivation of Sparse Basis Matrix ψ from a Gaussian Kernel

As the mentioned earlier, the method used for the deblurring involves segmentation of the image into 7x7 boxes. Thus, for an image with height h and width w pixel, the maximum value of the total number of boxes is given by:

$$Max. value\ of\ the\ total\ no. of\ boxes = (\frac{w}{7} + 1)(\frac{h}{7} + 1) \quad (8)$$

When a single 7x7 box is convolved with a 7X7 Gaussian kernel, there will be a resulting 49 set of equations with 49 unknown pixel values. These equations constitute the ψ matrix. Also, when a 7x7 box is convolved with a 7x7 Gaussian kernel only the central pixel at (3,3) is covered completely by the Gaussian kernel. Because of this, the convolution result will not be accurate at the rest of the pixels. Such effect is also observed when doing convolution at the edge of any image. For a convolution by a Gaussian kernel for example, this results in blackening of the image at the edges instead of whitening it. There are different existing methods that can be used to correct this such as wrapping or mirroring an image, ignoring edge pixels or duplicating edge pixels [23].

For the image convolution of 7x7 boxes at hand, ignoring the edge pixels cannot be an option because this will mean ignoring the whole image. But mirroring the image has been found a better option because symmetry is found frequently in nature and consequently in most pictures. After applying mirroring and doing the convolution of the 49 pixels of single box a normalized 49x49 ψ matrix will result.

2.4 Derivation of Measurement Matrix – M and Matrix – ϕ

The measurement matrix is mxn in size where $m<n$. The task of the measurement matrix is to reduce the n number of equations to m equations and it can be of different compositions. After doing the required computations and tests, finally the measurement matrix has been designed in such a way that it selects the even numbered rows from the main set of 49 equations. This measurement matrix is found to give the lowest coherence in the matrix ϕ which in turn gives favourable results in compressive sensing. Then, the ϕ matrix is obtained by using the relation

$$\phi = M\psi. \quad (9)$$

2.5 Selection of a Reconstruction Algorithm

There are different algorithms of compressive sensing that can be applied to a given problem. Here, the ϕ matrix which is derived from the image's convolution with the 2D-Gaussian kernel consists of coefficients that are all less than 1. As a result, the norm-2 of the matrix becomes less than 1 which makes it suitable to apply the Iterative Hard Thresholding algorithm of compressive sensing. This is because the Iterative Hard Thresholding algorithm is stated to converge whenever the norm-2 of the matrix ϕ is less than 1 [18]. For the ϕ matrix that has been used here, the second norm is found to be: $\|\phi\|_2 = 0.7649$.

2.6 Computation of Parameters Required for Use in Iterative Hard Thresholding Algorithm

The Iterative Hard Thresholding algorithm is listed below [18].

Listing.1 The Hard Iterative Thresholding Algorithm

Input

- s the sparsity of X

-y ε R^m and the matrix ϕ ε R^{mxn}

Output:

- *X' such that y = ϕx'*

1. $X^{(0)} = 0$

2. for i=1,…do

3. $X^{(i)} = Hs(X^{(i-1)} + \phi^T(y - \phi X^{(i-1)}))$

4. end for

5. $X' = X^{(i)}$

*where Hs is hard thresholding function which sets all components of the vector **x** to zero except the s largest components.*

$$\boldsymbol{Hs}(\boldsymbol{x}) = \begin{cases} x_i, & x_i \geq \varepsilon \\ 0, & x_i < \varepsilon \end{cases},$$

ε is the s-largest component of X

Before applying the Iterative Hard Thresholding algorithm some parameters need to be predetermined.

Setting m where m is the number of rows of the matrix ϕ(mxn): The integer *m* is the number of rows of the matrix ϕ. For each 7x7 box of the image, 49 equations were derived making *n=49*. Compressive sensing uses underdetermined system of equations where *m<n*. In order minimize the computational complexity of the Iterative Hard Thresholding algorithm m should be as much small as possible. This is because the Iterative Hard Thresholding algorithm has computational complexity of O(*mn*) per iteration [24]. However, a balance should be kept in the minimizing of *m* which otherwise will compromise the restoration of the image. Keeping that in consideration, *m* is selected to be, *m = 16* for initial trial.

Setting the sparsity value s: According to Equation (4), *s* is calculated as:

$$s = \frac{m}{2\log n} = \frac{16}{2\log 49} = 4.733 \cong 5 \qquad (10)$$

Setting the upper bound ε of the thresholding function: *ε* is the s-largest element of X. It is also possible that *ε* can be selected randomly whenever there is no unique set of *s* number of elements which are the largest in the vector X [18]. The pixel values in an image range from 0-255. And most images contain much larger number of pixels than 255. As a result, most of the time there will not be found a unique set of *s* number of elements which are the largest pixel values in an image. Because of this, *ε* has been set randomly. Thus, *ε* is made to be equal to the sparsity value *s* giving *ε = 5*.

Computing theoretical maximum number of iterations k^*: The theoretical maximum number of iterations can be computed using Equations (11) and (12). And it holds true when the matrix ϕ has modified Restricted Isometric Property (RIP) of order 3s, $\beta_{3s} < 1/8$, where $\beta_s = 1 - \frac{1-\delta_s}{1+\delta_s}$. The calculation of RIP is a computationally intensive process and this is stated in [24]. Due to this, the Restricted Isometry Property (RIP), δ_s, has not been computed for the matrix ϕ . As it is mentioned before, the precondition for convergence of the Iterative Hard Thresholding algorithm is satisfied with the norm-2 of the matrix ϕ being less than 1. But with the RIP value of the matrix ϕ being unknown, the theoretical maximum number of iterations may not hold true. As a result, the theoretical maximum number of iterations is computed and then taken as the starting number of iterations with which the Iterative Hard Thresholding algorithm is applied.

The theoretical maximum number of iterations, k^*, is computed as follows.

Given, $Y = \phi X + e$ (where e is the noise vector in the blurred image) and X' as the k^{th} approximation,

$$k^* = \left\lceil \log_2\left(\frac{\|X'\|_2}{\varepsilon_s}\right)\right\rceil \quad (11)$$

$$\varepsilon_s = \|X - X'\|_2 + \frac{1}{\sqrt{s}}\|X - X'\|_1 + \|e\|_2 \quad (12)$$

$\|X'\|_2$ can have different values depending on the pixel color. Here, the maximum value of $\|X'\|_2$ is computed with the maximum value a pixel can have which is 255. X is n-dimensional with *n=49*. The norm-2 and norm-1 of the error vector, $\|X - X'\|$, is computed assuming a maximum of 1 pixel difference between the actual solution and the approximation vector. The noise is assumed to be zero giving $\|e\|_2 = 0$. With these values set, the theoretical maximum number of iterations becomes: $k^* = 6$

Initial approximation value: The initial approximation vector has been set to have a value of white pixel which is 255 i.e. $X^{(0)} = 255$.

3. RESULTS AND DISCUSSION

3.1 Specification of the Inputs Used for Deblurring

We used 24 bit bitmap images as inputs for testing our proposed work. Thus, the algorithm has been applied three times in each image for the R(red), G(green), B(blue) array of pixels.

The blurred images that are used as inputs in the deblurring process have been convolved box-wise. This means that when performing convolutions on the original images, the images are segmented into 7x7 pixel boxes and the convolution is done iteratively on each box separately. And the mirroring of pixels near the borders of each box has been done during convolution. This is not the natural way of performing convolution. It has been done in order to match the input blurred image with the model used for deblurring.

3.2 Applying IHT to Images and Results obtained

When IHT was applied to an image with the initial parameters computed in Sec. III an acceptable outcome was not obtained. In the resulting image, each box has not been restored sufficiently and because of that grids have been formed all over the image.

In order to improve the deblurring, m has been varied to different values and good results were obtained for *m*=25 by selecting even number of rows from ψ. This is because the ϕ matrix has the lowest correlation when such a selection is done. Table 1 gives few samples of correlation values of ϕ for different types of measuring matrix M where $\phi = M\psi$.

Table 1 Correlation values of different values of the measurement matrix

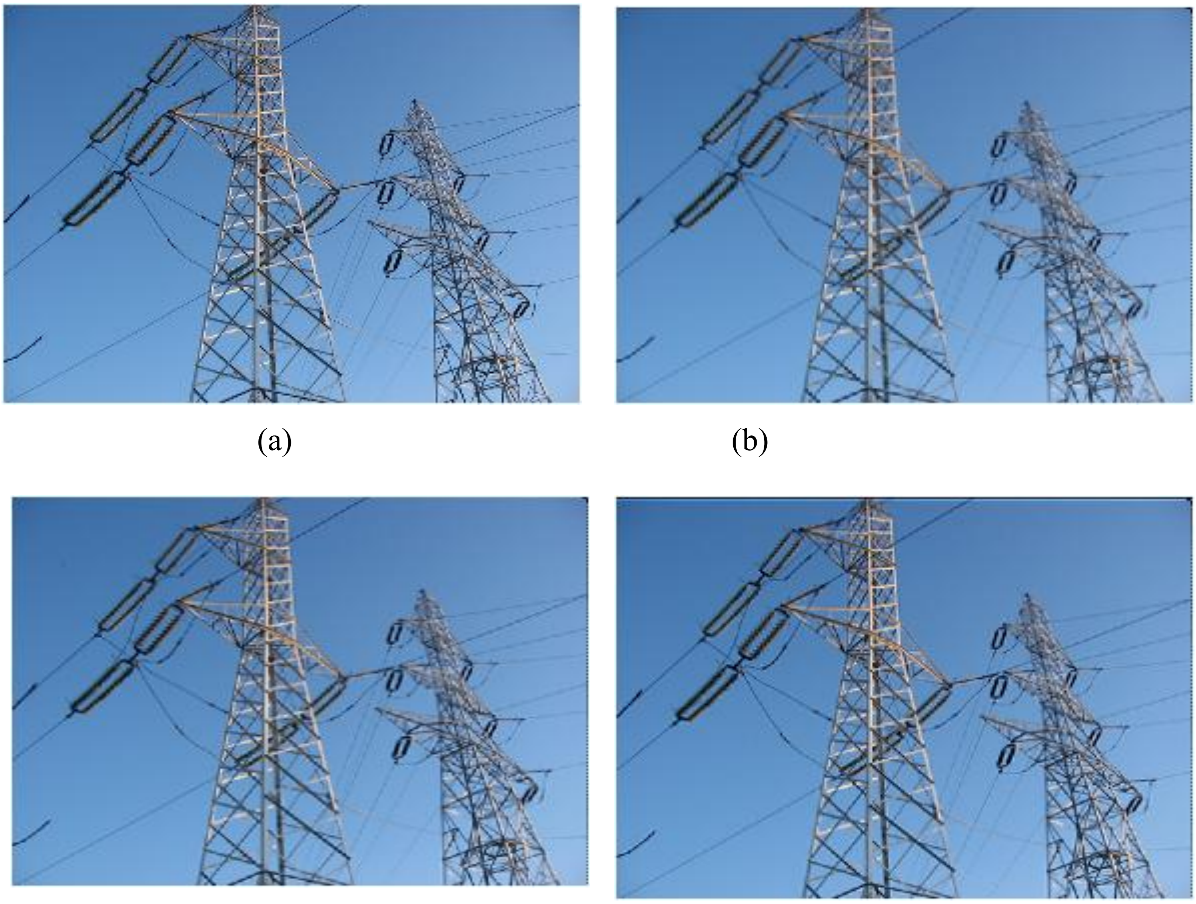
Type of Measurement matrix(M)	Value of m	Correlation of ϕ
Selects rows of ψ which are multiples of 3	16	0.979235
Selects rows of ψ which are even	25	0.953705
Selects rows of ψ randomly	32	0.975379

Iterative Hard Thresholding was applied to the convolved Electric_Lines image displayed in Figure 1 with the improved parameters shown in Table 2.

Table 2 Improved Parameters

No of rows of ϕ(m)	Upper bound of thresholding(ε)	No of Iteration(k)
25	5	6

In the deblurring result for the Electric_Lines image using *m=25* and the other parameters unchanged, an improvement of only 0.4% in Green pixels and 1 % in Blue pixels was obtained while the Red pixels deteriorated by 0.6%. Thus, the number of iterations was increased successively and the image was deblurred in better percentages. In the process, good results were obtained for $k = 45$. Results for $k = 12$ and $k = 45$ are shown in Figure 1c and Figure 1d.



(a) (b)

(c) (d)

Figure 1 (a) The original Electric_Lines image, (b) Box wise Gaussian convolved image, (c) Deblurring result for m=25, k=12, (d) Deblurring result for m=25, k=45

Results of IHT applied to a box-wise Gaussian convolved flower image are shown in Figures 2 (a to d).

(a) (b)

(c) (d)

Figure 2 (a) Original Flower Image, (b) Box-wise convolved Flower Image, (c) Deblurring result for k = 12, (d) Deblurring result for k = 45

3.3 Mean Squared Error (MSE) of the Deblurred Images

Mean Squared Error (MSE) of the deblurred images is shown in Table 3. The results show that IHT has deblurred the images to a measurable degree. In both images, the deblurring is negligible or negative at the theoretical maximum iteration $k=6$. As it has been stated earlier, this theoretical maximum iteration has been used as starting number of iterations since the RIP of the matrix ɸ is unknown. Thus, when the number iterations is increased to $k=12$, there is a considerable improvement in the deblurring ranging from 4%-30%. At iteration number $k=45$, the images show a much better deblurring result ranging from 10%-40%. During the test, it has been observed that the number of iterations cannot be increased from a certain upper bound beyond which the image will be degraded.

Table 3 MSE of the deblurred images

Image	MSE of Blurred image (RGB)	No. of Iterations-k	MSE of Deblurred image	Improvement
Electric_Lines	(R) 374.6 (G) 343.7 (B) 341.2	6	(R) 376.9	-0.6%
			(G) 342.0	0.4%
			(B) 337.9	1%
		12	(R) 349.5	6.7%
			(G) 316.0	8%
			(B) 308.8	9.4%
		45	(R) 317.2	15.5%
			(G) 289.1	15.9%
			(B) 272.0	20.3%
Flower	(R) 175.3 (G) 128.5 (B) 175.9	6	(R) 168.4	3.9%
			(G) 130.2	-1.3%
			(B) 169.7	3.5%
		12	(R) 123.5	30%
			(G) 113.5	11.7%
			(B) 133.0	22.4%
		45	(R) 105.2	40%
			(G) 115.3	10.3%
			(B) 120.7	31.4%

3.4 Computational Complexity of the Deblurring Method

The computational complexity analysis given below refers to both the time and storage requirement. The IHT algorithm has computational complexity of O(mn) per iteration [24]. For k number of iterations, the total computational complexity becomes O(kmn). The computational complexity of the deblurring method followed involves the cost of evaluating IHT at each box multiplied by the total number of boxes in the image. In addition, it includes the cost of evaluating ψ and ϕ matrices and the segmenting overhead.

For the analysis below,

n - represents the total number of pixels in a segmented box for this case a 7x7 box making *n=49*

m - represents the number of rows in the ∅ matrix making *m=25*

k - represents the number of iterations

Computational complexity of the evaluating the ψ matrix: Evaluating the ψ matrix includes representing the convolution of a 7x7 pixel box of the image with the 7x7 2D Gaussian kernel by 49x49 matrix. This process includes three steps. The first one is initial representation of the convolution coefficients and the 7x7 box pixel indexes by nxn matrix whose elements need to be ordered further. This step has computational complexity of $O(n^2)$. The next step is ordering of each row of the nxn matrix based on pixel indexes using a sorting algorithm. This step gives the initial ψ matirx before mirroring. The ordering or sorting algortihm used for this second step is insertion sorting. Insertion sort has computational complexity $O(n^2)$ and when it is applied for each of the n rows of the unordered ψ matrix, the resulting total computational complexity is $O(n^3)$**.** Finally, the ψ matrix is mirrored to account for the incomplete convolution near the edges of the box. This final step has computational complexity of $O(nxn) = O(n^2)$**.**

Thus, the total computational complexity of evaluating the ψ matrix is the sum of the complexities of the above processes and it is equivalent to $O(n^3)$.

Computational complexity of the evaluating the ∅ matrix: The ∅ matrix is given by $\emptyset = M\psi$ where M is mxn and ψ is nxn. This matrix multiplication has $O(mxnxn) = O(mn^2)$ computational complexity.

Computational complexity of applying IHT on a single 7x7 box: As stated above, Iterative Hard Thresholding, IHT, has computational complexity of *O(kmn).* In the box-wise approach, there is a cost of segmenting the image at each iteration which increases the computational complexity by a factor of 49. This is due to the cost of fetching an array of length 49 pixels from the total collection of the image's pixels at each iteration. Thus, the total computational complexity becomes *O(kmnSO)* where *SO = n = 49* is the segmenting overhead. Computational complexity of applying IHT a single 7x7 box $= O(kmnSO) = O(kmn^2)$

Computational complexity of applying IHT on the whole image: As it has been stated earlier, the total number of boxes in a 7x7 segmented image is given by:

$$Total\ no. of\ boxes = (\frac{w}{7} + 1)(\frac{h}{7} + 1) \cong \left(\frac{w}{7}\right)(\frac{h}{7})$$

Where w - width of the image

h - height of the image

Computational complexity of applying IHT on the whole image =

$$O\left(kmn^2\left(\frac{w}{7}\right)\left(\frac{h}{7}\right)\right) = O(kmn^2\left(\frac{wh}{49}\right)) = O(kmn(wh))$$

The total Computational complexity of deblurring the whole image: The total computational complexity of deblurring the whole image is approximately the largest of the computational costs stated above. The largest computational cost is when applying IHT on the whole image. Thus, the total computational complexity of deblurring the whole image is $O(kmn(wh))$.

Computational Complexity of Applying IHT using non-Box-wise Method

If the Iterative Hard Thresholding algorithm had been used without segmenting the image into boxes, the expected computational complexity would have been as follows. In the case where non-box-wise method is used, the ψ matrix has dimensions *sxs* where *s = wh.*

For the analysis below

n - represents the total number of rows in the ψ matrix making *n = s = wh.*

m - represents the number of rows in the ∅ matrix making $m = \frac{n}{2} = \frac{wh}{2}$ assuming that half of the equations are selected.

k - represents the number of iterations as before.

Computational complexity of the evaluating the ψ matrix: Evaluating this computational complexity comprises the copying into, ordering and applying mirroring to the ψ matrix which has dimension *sxs* where *s=wh.* The computational complexity of these processes amounts to $O((wh)^3)$. This value is obtained by replacing the box-size with the image size, *n=s=wh,* in the previous derivation of the computational complexity of the ψ matrix in the box-wise deblurring method.

Computational complexity of the evaluating the ∅ matrix: As stated earlier, this computational complexity has order of multiplying the M*(mxn)* matrix by the $\psi(nxn)$ matrix to give the ∅*(mxn)* matrix. By replacing the m and n values are given above for the non-box wise deblurring case, this step will have a computational complexity of $O\left(\frac{(wh)^3}{2}\right) = O((wh)^3)$.

Computational complexity of applying IHT on the whole image: Substituting the values k, n and m for the non-box wise method stated above into the computational complexity of applying IHT to the box-wise method gives:

Computational complexity of IHT without segmenting into boxes = $O(knm)$=$O(k\frac{(wh)^2}{2})$ = $O(k(wh)^2)$

The total computational complexity of deblurring the whole image: The total computational complexity of deblurring the whole image by using non-box wise method is the largest computational complexity of the above steps. And this largest value is obtained when evaluating the ψ matrix giving computational complexity of $O((wh)^3)$.

3.5 Comparison of the Box-wise Method of Deblurring with Other Methods

As computed earlier, the box wise deblurring method has computational complexity of $O(kmn(wh)$. By substituting the values of *k*=45, *m*=25 and *n*=49, the computational complexity becomes $(55125(wh))$. A typical printing size of an image has a resolution of 540 x 360 pixels giving a total pixel size of *wh*=194,900[25]. Thus, for such an average sized image, it can be seen that the computational complexity is equivalent to $O((wh)^2)$. Applying IHT for the whole image, without using box-wise approach has computational complexity of $O((wh)^3)$. Directly inverting the ψ matrix by Gauss Jordan method to get the original image has computational complexity of $O((wh)^3)$ [26] while the Lucy-Richardson algorithm has computational complexity of $O(k(wh)^3)$ [27], [28].

3.6 Limitations of our work

When normally (non-box-wise) convolved images were attempted to be deblurred with the box-wise deblurring method, the process resulted in no deblurring at any number of iterations. The reason behind this is the mirroring used in the 7x7 boxes of the images while deriving the deblurring model. Here, the box size and the kernel size used are the same resulting in only the central pixel to be convolved in the normal way without mirroring. This has resulted in too much approximation in the box-wise method so that a normally convolved image couldn't be deblurred using the given way.

4. CONCLUSIONS

In this paper, a box-wise method of deblurring images using compressive sensing has been introduced. When applied to box-wise convolved images, this box-wise deblurring method has been found to be computationally more efficient than using the non-box wise counterpart or using the direct matrix inversion method. The method also exhibited a better computational efficiency than the well-known Lucy-Richardson deblurring.

However, when the box-wise deblurring method was applied to normally convolved images the results were not desirable. As it is stated earlier, when the convolution is done box-wise, mirroring is applied at those pixel points which the kernel cannot cover completely. When the box size and the kernel size are the same, in the given case 7x7, only the central pixel of the box gets convolved in a normal way and for the rest of the 48 pixel mirroring must be done. That means, only 2% of the box is convolved in the conventional way which results in the mismatch of the deblurring model and normally convolved images.

To minimize the approximations resulting from mirroring, the box size can be increased. But this will be at the cost of increasing the computational complexity which is given by $O(kmn(wh)$ where n is the box size. Finding an optimum design which minimizes the number of iterations, k, and the value of m can compensate for the increase in computational complexity caused by increasing the box size. In doing these, the deblurring method could be applicable to normally convolved images without its computational complexity being compromised.

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

ACKNOWLEGMENTS

The authors are deeply grateful to God the father, the Son and the Holy Spirit.

REFERENCES

[1] S. J. Reeves, "Image Restoration: Fundamentals of Image Restoration", in mic Press Library in Signal *Processing,* Elseviear, 2014, pp. 165-192.

[2] D. Singh, and . K. Sahu, R "A Survey on Various Image Deblurring Techniques", *International Journal of Advanced Research in computer and communication engineering,* vol. 2, no. 12, 2013.

[3] D. L. Donoho, "Compressive Sensing", I*EEE Transactions on Information Theory,* vol. 52, no. 4, pp. 1289-1306, 2006.

[4] E. J. Candes, and T. Tao, "Decoding by Linear Programming", *IEEE Transactions on Information Theory,* vol. 51, no. 12, pp. 4203-4215, 2005.

[5] B Amizic, L. Spinoulas, R. Molina, and A. K. Katsaggeolos, , "Compressive Blind Image Deconvolution", *IEEE Transactions on Image Processing,* vol. 22, no. 10, pp. 3994-4006, 2013.

[6] E. J. Candes, J. Romberg, and T. Tao, "Robust Uncertainty Principle: Exact Signal Reconstruction from Highly Incomplete Frequency Information", *IEEE Transactions on Information Theory,* vol. 52, no. 4, pp. 489-509, 2006.

[7] D. L. Donoho, "For Most Large Underdetermined Systems of Linear

Equations the Minimal 1-norm Solution is also the Sparsest Solution", vol. 59, no. 6, pp. 797-829, 2006.

[8] E. J. Candes, "The Restricted Isometry Property and its Implications for Compressed Sensing", 2008.

[9] J. A. Tropp, "Just Relax: Convex Programming Methods for Identifying Sparse Signals in Noise", vol. 52, no. 3, pp. 1030-1051, 2006.

[10] M. A. Davenport, M. F. Duarte, Y. C. Eldar, and G. Kutyniok, "Introduction to Compressive Sensing" [Online]. Available: webee.technion.ac.il.

[11] H. Cheng, "Modeling and Learning Visual Recongintion Theory Algorithms and Applications", 2015. [Online]. Available: www.Springer.com.

[12] D. L. Donoho, and J. Tanner, "Counting Faces of Random Projected Polytopes when the Projections Radically Lowers Dimensions", *Journal of the American Mathematical Society,* vol. 22, no. 1, 2006.

[13] J. A. Tropp, "Greed is Good: Algorithmic Results for Sparse Approximation", vol. 50, no. 10, pp. 2231-2242, 2004.

[14] J. A. Tropp, and A. C. Gilbert, "Signal Recovery from Random Measurement via Orthogonal Matching Pursuit", vol. 53, no. 12, 2007.

[15] S. G. Mallat, and Z. Zhang, "Matching Pursuits with Time-Frequency Dictionaries", *IEEE Transactions on Signal Processing,* 1993.

[16] S. G. Mallat, G. Davis, and Z. Zhang, "Adaptive Time Frequency Decompostions with Matching Pursuits", *SPIE Journal of Optical Engineering,* vol. 33, pp. 2183-2191, 1994.

[17] D. Needell, and J. A.Tropp, "CoSamp: Iterative signal recovery from incomplete and inaccurate samples", *Applied and Computational Harmonic Analysis,* vol. 26, no. 3, pp. 301-321, 2009.

[18] T. Blumensath, and M. E. Davis, "Iterative Thresholding for Sparse Approximations", *Journal of Fourier Analysis and Applications,* vol. 14, no. 5, pp. 629-654, 2008.

[19] A. Tavakoli, and A. Pourmohammad, "Image Denoising Based on Compressed Sensing", *International Journal of Computer Theory and Engineering,* vol. 4, no. 2, 2012.

[20] J. Yu, Z. Chang, C. Xiao and W. Sun, "Blind Image Deblurring Based on Sparse Representation and Structural Self-similarity", in 42th *International Conference on Accoustics,* Speech and Signal Processing, 2017.

[21] C. A. Metzler, A. Maleki, and R. Baraniuk, "From Denoising to Compressed Sensing", *IEEE Transactions on Information Theory,* 2014.

[22] W. Kang, E. Lee, S. Kim, D. Seo, and J. Paik, "Compressive Sensing Based Image Denoising Using Adaptive Multiple Samplings and Reconstruction Error Control", Proceedings of SPIE- The International Society for Optical Engineering, vol. 8365, 2012.

[23] Ludwig, J., *"Satellite Digital Image Analysis"*, [Online]. Available: web.pdx.edu.

[24] Blumensath, T. and Davis M. E., *"Iterative Hard Thresholding for Compressed Sensing"*, Applied and Computational Harmonic Analysis, vol. 27, no. 3, pp. 256-274, 2009.

[25] "*Image Resolution and DPI Requirement"*, Walgreen Co., 24 07 2017. [Online]. Available: http://wagcco.secure.force.com. [Accessed 28 09 2021].

[26] Farebrother, R., *"Linear Least Square Computations"*, STATISTICS, 1988.

[27] Lucy, L. B., "*An Iterative Technique for the Rectification of Observed Distributions"*, Astronomical Journal, vol. 79, no. 6, 1974.

[28] Richardson, W. H., "*Bayesian-Based Iterative Method of Image Restoration*", Journal of the Optical Society of America, vol. 62, no. 1, 1972.