Blind Deconvolution

Mode: Application

1st Chaman Agrawal
160205

2nd Maab Zia Khan
160406

***Abstract*—Blind Deconvolution is the problem of recovering two unknown signals w and x from their circular convolution with or without noise. This problem arises in many areas from Image processing, signal processing to astronomy and wireless communication. One of the major challenges in the Blind Deconvolution problem is its non-convex nature. It has been observed that this problem suffers from a large number of local minimas when posed as an optimization problem.**
In our paper we have implemented four different algorithms to solve this problem and have tried to compare their performance under various settings. Two of the algorithms are studied over signal vectors while the other two are specifically studied for the Image Deblurring problem.

I. Introduction

In signal processing whenever the transmitter sends a signal $\boldsymbol{x}$ through a channel with channel response $\boldsymbol{w}$ with noise $\boldsymbol{n}$, then the received signal $\boldsymbol{y}$ will be of the form

$$\boldsymbol{y} = \boldsymbol{w} * \boldsymbol{x} + \boldsymbol{n}$$

Similar analogy can be drawn in the field of image processing where $\boldsymbol{x}$ is the true image, $\boldsymbol{w}$ is a blur kernel and $\boldsymbol{y}$ is the obtained image. So, the problem to recover the $\boldsymbol{x}$ and $\boldsymbol{w}$ from $\boldsymbol{y}$ is called the Deconvolution problem. There are two types of this problem depending on whether the channel response / blur kernel $\boldsymbol{w}$ is known *i.e.* non-Blind Deconvolution or unknown *i.e.* Blind Deconvolution.
Blind Deconvolution is a highly ill-posed problem because it is not only non-convex but also has non-unique optimal solutions. To better frame the Blind Deconvolution problem, several assumptions are made, each leading to different approaches and applications in the real world. The algorithms we have considered also use different assumptions like known subspaces on $\boldsymbol{w}$ and $\boldsymbol{x}$, coherence between $\boldsymbol{w}$ and $\boldsymbol{x}$, priors etc.

II. Algorithms

We have implemented four algorithms for the Blind Deconvolution problem. The first two algorithms are designed of general signal vectors while the later two algorithms are specifically for the Image deblurring problem. These are the algorithms implemented:

- Convex Blind Deconvolution (CBD) [1]
- Non-convex Blind Deconvolution (NCBD) [2]
- Provably Robust Image Deconvolution Algorithm (PRIDA) [3]
- Graph based Blind Deblurring (GBD) [4]

Algorithm-1: Convex Blind Deconvolution

Consider the original problem:

$$\boldsymbol{y} = \boldsymbol{w} * \boldsymbol{x} + \boldsymbol{n} \tag{1}$$

Assume that the length L signals $\boldsymbol{w}$ and $\boldsymbol{x}$ are from known subspaces of $\mathbb{R}^L$ with dimensions K and N respectively. So,

$$\boldsymbol{w} = \boldsymbol{B}\boldsymbol{h}, \quad \boldsymbol{h} \in \mathbb{R}^K,\ \boldsymbol{B} \in \mathbb{R}^{L\times K}$$
$$\boldsymbol{x} = \boldsymbol{C}\boldsymbol{m}, \quad \boldsymbol{m} \in \mathbb{R}^N,\ \boldsymbol{C} \in \mathbb{R}^{L\times N}$$

Equation (1) can be rewritten as:

$$\boldsymbol{y} = \begin{bmatrix} \mathrm{circ}(C_1)\boldsymbol{B} & \cdots & \mathrm{circ}(C_N)\boldsymbol{B} \end{bmatrix} \begin{bmatrix} m(1)\boldsymbol{h} \\ m(2)\boldsymbol{h} \\ \vdots \\ m(N)\boldsymbol{h} \end{bmatrix}. \tag{2}$$

where $\boldsymbol{C_n}$ are columns of $\boldsymbol{C}$ and $\mathrm{circ}(C_n)$ is an $L \times L$ matrix for circular convolution action with vector C_n.

Using L-point normalized discrete Fourier transform (DFT) matrix $\boldsymbol{F}$, equation (2) becomes

$$\Delta_n = \mathrm{diag}\left(\sqrt{L}\hat{C}_n\right)$$

$$\hat{\boldsymbol{y}} = \boldsymbol{F}\boldsymbol{y} = \begin{bmatrix} \Delta_1\hat{\boldsymbol{B}} & \Delta_2\hat{\boldsymbol{B}} & \cdots & \Delta_N\hat{\boldsymbol{B}} \end{bmatrix} \begin{bmatrix} m(1)\boldsymbol{h} \\ m(2)\boldsymbol{h} \\ \vdots \\ m(N)\boldsymbol{h} \end{bmatrix} \tag{3}$$

where $\hat{\boldsymbol{B}} = \boldsymbol{F}\boldsymbol{B}$, $\hat{\boldsymbol{C}} = \boldsymbol{F}\boldsymbol{C}$. This can be simplified to:

$$\hat{\boldsymbol{y}}(\ell) = \mathrm{trace}\left(\boldsymbol{A}_\ell^*\left(\boldsymbol{h}\boldsymbol{m}^*\right)\right), \quad \text{where} \quad \boldsymbol{A}_\ell = \hat{\boldsymbol{b}}_\ell\hat{\boldsymbol{c}}_\ell^*$$

While $\boldsymbol{y}$ is a nonlinear combination of $\boldsymbol{h}$ and $\boldsymbol{m}$, equation (3) shows that $\boldsymbol{y}$ is a linear combination of their product $\boldsymbol{X_0} = \boldsymbol{h}\boldsymbol{m}^*$. And since $\boldsymbol{X_0}$ is rank-1, it is recoverable even for $\boldsymbol{L} << \boldsymbol{NK}$. So, the non-linear deconvolution problem had been changed to a linear inverse problem over the set of rank-1 matrices.
Further, the author suggests to use the following least squares problem to choose from the multiple possible solutions of the above problem:

$$\min_{u,v} \|u\|_2^2 + \|v\|_2^2$$
subject to
$$\hat{y}(\ell) = \langle \hat{c}_\ell, u\rangle \langle v, \hat{b}_\ell\rangle, \quad \ell = 1, \dots, L$$

The above non-convex quadratic optimization problem is simplified to the following nuclear norm convex problem using the 'dual-dual' relaxation heuristic as presented in [1].

$$\min_{W_1, W_2, X} \frac{1}{2}\text{trace}(W_1) + \frac{1}{2}\text{trace}(W_2)$$
$$\text{subject to } \begin{bmatrix} W_1 & X \\ X^* & W_2 \end{bmatrix} \succeq 0$$
$$\hat{y} = \mathcal{A}(X)$$

or simply as:

$$\min \quad \|X\|_*$$
$$\text{subject to} \quad \hat{y} = \mathcal{A}(X)$$

The paper supports the effectiveness of the algorithm for relatively large K and N when B is incoherent (coherence measured as given in [1]) and orthonormal, and C from a random from an isotropic distribution with rows of $\hat{c}$ being independent, along with the assumption that w is time-limited. Finally, the paper provides theoretical guarantee for the convergence of the proposed algorithm and it's time complexity.

Algorithm 1 Convex Blind Deconvolution

```
1: $A_\ell = \hat{b}_\ell \hat{c}_\ell^*$
2: $\min \quad \|X\|_*$ subject to $\quad \hat{y} = \mathcal{A}(X)$
3: $[U, S, V] = \text{svd}(X)$
4: $\hat{w} = B * \hat{h}$
5: $\hat{x} = C * \hat{m}$
```

Algorithm-2: Non-Convex Blind Deconvolution

The overall setting in this algorithm is similar to [1], one key difference here is that here h and x are complex valued vectors compared to real vectors in Algorithm-1. So,

$$y = f * g + n$$
$$g = Cx, \quad x \in \mathbb{R}^N$$
$$Ff = Bh$$

where F is the L-point unitary DFT matrix. and n is additive white complex Gaussian noise.
Moving to the frequency domain, we get

$$\sqrt{L}Fy = \text{diag}(\sqrt{L}Ff)(\sqrt{L}Fg) + \sqrt{L}Fn$$
$$\frac{1}{\sqrt{L}}\hat{y} = \text{diag}(Bh)\overline{Ax} + \frac{1}{\sqrt{L}}Fn$$
$$\text{where} \quad \overline{A} = FC$$

We define two linear operators:

$$\mathcal{A}(Z) = \left\{b_l^* Z a_l\right\}_{l=1}^{L}, \quad \text{and } \mathcal{A}^*(z) = \sum_{l=1}^{L} z_l b_l a_l^*$$

where a_l,b_l denotes the l^{th} column of A^* and B^* respectively.

We arrive at the problem:

$$\min_{(h,x)} F(h, x)$$
where
$$F(h, x) := \|\text{diag}(Bh)\overline{Ax} - y\|^2$$
$$= \left\|\mathcal{A}\left(hx^* - h_0 x_0^*\right) - e\right\|^2$$
$$e = \frac{1}{\sqrt{L}}Fn$$

The above problem is a challenging nonlinear least square optimization problem which is highly non-convex. The paper finally proposes a method to find a good initial guess for h and x and then a nice gradient method to arrive at the optimal solution without getting stuck at the local minimas.
The paper defines the following three important sets:

$$\mathcal{N}_{d_0} := \left\{(h, x) : \|h\| \le 2\sqrt{d_0}, \|x\| \le 2\sqrt{d_0}\right\}$$
$$\mathcal{N}_\mu := \left\{h : \sqrt{L}\|Bh\|_\infty \le 4\sqrt{d_0}\mu\right\}$$
$$\mathcal{N}_e := \left\{(h, x) : \left\|hx^* - h_0 x_0^*\right\|_F \le \varepsilon d_0\right\}$$

where

$$\mu^2 \ge \frac{L\,\|Bh_0\|_\infty^2}{\|h_0\|^2}, \quad d_0 = \|h_0\|\,\|x_0\|$$

The algorithm consists of two parts: Finding an initial guess inside the region $\mathcal{N}_{d_0} \cap \mathcal{N}_\mu \cap \mathcal{N}_\varepsilon$. and then carefully applying a regularized gradient descent algorithm that will ensure that all the iterations remain inside that region. To ensure this, we add the following regularization term to $F(h,x)$:

$$G(h, x) = \rho\left[G_0\left(\frac{\|h\|^2}{2d}\right) + G_0\left(\frac{\|x\|^2}{2d}\right) + \sum_{l=1}^{L} G_0\left(\frac{L\left|b_l^* h\right|^2}{8d\mu^2}\right)\right]$$

From [2]:

Algorithm 2 Non-Convex Blind Deconvolution: Part-1

```
1: Compute $\mathcal{A}^*(y)$.
2: Find the leading singular value, left and right singular
   vectors of $\mathcal{A}^*(y)$, denoted by $d$, $\hat{h_0}$ and $\hat{x_0}$ respectively.
3: $u_0 := \text{argmin}_z \left\|z - \sqrt{d}\hat{h}_0\right\|^2$ .
   subject to $\sqrt{L}\|Bz\|_\infty \le 2\sqrt{d}\mu$
4: $v_0 := \sqrt{d}\hat{x}_0$
```

Algorithm 3 Non-Convex Blind Deconvolution: Part-2

```
1: Initialization: Obtain $(u_0, v_0)$ via Part-1
2: for $t = 1, 2, ...$ do
3:     $u_t = u_{t-1} - \eta \nabla \widetilde{F}_h(u_{t-1}, v_{t-1})$
4:     $v_t = v_{t-1} - \eta \nabla \widetilde{F}_x(u_{t-1}, v_{t-1})$
```

Algorithm-3: PRIDA

Let $\boldsymbol{f} \in \mathbb{R}^n$ be the vectorized sharp image and $\boldsymbol{k} \in \mathbb{R}^s$ be the blur kernel to be estimated. Let $\boldsymbol{b} \in \mathbb{R}^n$ be the given vectorized blurry image, then

$$b = f * k + \alpha \tag{4}$$

where α denotes the independent noise vector at each pixel. To estimate $\boldsymbol{f}$, $\boldsymbol{k}$ we can solve the following log-likelihood optimization problem:

$$\min_{\boldsymbol{f},\boldsymbol{k}} \|\boldsymbol{f} * \boldsymbol{k} - \boldsymbol{b}\|_2^2 \tag{5}$$

Assuming the kernel $\boldsymbol{k}$ to be non-negative and $\boldsymbol{f}$ be non-negative since it represents an image. To find solutions to (5), the paper suggests a Total Variation regularizer (TV), so we get the following non-convex problem :

$$\min_{f,k\in\Delta} \mathcal{L}(f,k) := \|f * k - b\|_2^2 + \lambda \|f\|_{TV} \tag{6}$$

where

$\lambda \geq 0$ is a tunable regularization parameter,

Δ is the probability simplex $:= \{x \geq 0 : 1^T x = 1\}$

$\|f\|_{TV}^p := \left(\left(\|\nabla_i f\|_p + \left\|\nabla_j f\right\|_p\right)^p, \quad p = 2\right.$

To solve (6), the paper presents a novel algorithm *PRIDA*, which is similar to the Mirror Descent algorithm (MD) in Convex Optimization. A major difference between the two algorithms is that unlike *MD*, in *PRIDA* the step size is chosen independently for each coordinate.

Algorithm 4 PRIDA

```
1: Pick a starting point $k^0 \in \Delta, f^0 \in [0,1]^n$.
2: for $t = 1, 2, ... T$ do
3:     $\left(g_f^t, g_k^t\right) \leftarrow \left(\nabla \mathcal{L}_f\left(f^t, k^t\right), \nabla \mathcal{L}_k\left(f^t, k^t\right)\right)$
4:     $f^{t+1} \leftarrow f^t - \eta_f g_f^t$
5:     $\hat{k}_i^{t+1} \leftarrow k_i^t \exp\left(-\eta_{k_i}^t g_{k_i}^t\right)$
6:     $k^{t+1} \leftarrow \frac{\hat{k}^{t+1}}{\sum_{i=1}^s \hat{k}^{t+1}}$
```

Algorithm-4: Graph based Blind Deblurring

Consider the problem:

$$b = x * k + n$$

where b is the observed blurry image, x is the original sharp image, k is the blur kernel and n is the noise. Given b, we want to find the original image x and the blur kernel k.

The paper explores relation between graphs and blurred image, and proposes a new graph-based prior *RGTV*. We first compute a piece-wise smooth skeleton image that contains the strong gradients of the original image but not the minor details to estimate the blur kernel, then we find the target image using the estimated kernel and a non-blind deconvolution algorithm.

We will first define the graph model for images as given in the paper.

Graph: A graph $G(V, E, W)$ is a triplet consisting of a finite set V of N nodes (image pixels) and a finite set $E \subset V \times V$ of M edges. Each edge $(i, j) \in E$ is undirected with a corresponding weight $w_{i,j}$ which measures the similarity between nodes i and j. Here we compute the weights using a Gaussian kernel

$$[W]_{i,j} = w_{i,j} = \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{\sigma^2}\right)$$

where W is an *adjacency matrix* of size $N \times N$, x_i and x_j are the intensity values at pixels i and j of the image x, and σ is a parameter.

Laplacian Matrix: Given the adjacency matrix W, a combinatorial graph Laplacian matrix L is a symmetric matrix defined as:

$$L \triangleq diag(W.\mathbf{1}) - W$$

where $\mathbf{1}$ is a vector of all 1's.

Bi-modal Distribution: It means that the inter-pixel differences in an image patch are either very small or very large, *i.e.*, the patch is piece-wise smooth.

With the help of above definition, the *Reweighted Graph Total Variation* (RGTV) prior is defined as:

$$\begin{aligned} \|\mathbf{x}\|_{RGTV} &= \sum_{i\in\mathcal{V}} \|\text{diag}\left(\boldsymbol{W}_{i,\cdot}(\mathbf{x})\right) \nabla_i \mathbf{x}\|_1 \\ &= \sum_{i=1}^{N}\sum_{j=1}^{N} w_{i,j}\left(x_i, x_j\right)\left|x_j - x_i\right| \end{aligned}$$

where $\boldsymbol{W}_i \cdot (\mathbf{x})$ is the i-th row of $\boldsymbol{W}(\mathbf{x})$ and $w_{i,j}\left(x_i, x_j\right)$ is the (i, j) element of the adjacency matrix $\boldsymbol{W}$.

So, the problem we want to solve is:

$$\hat{x}, \hat{k} = \arg\min_{\boldsymbol{x},\boldsymbol{k}} \frac{1}{2}\|\boldsymbol{x} * \boldsymbol{k} - \boldsymbol{b}\|_2^2 + \beta\|\boldsymbol{x}\|_{RGTV} + \mu\|\boldsymbol{k}\|_2^2 \tag{7}$$

where β and μ are parameters.

The paper suggests a *coarse-to-fine* strategy to solve (7) where we work in a pyramidal structure by first down-sampling the blurry image and performing blind image blurring on it level-by-level. In each level, we estimate $\hat{\boldsymbol{k}}$ and $\hat{\boldsymbol{x}}$, and then up-sample $\hat{\boldsymbol{k}}$ as initial value for the next level.

Algorithm 5 Graph based Blind Deblurring

```
1: Input: Blurry image $\boldsymbol{b}$ and kernel size $h \times h$.
2: while not converge do
3:     Compute $\hat{\boldsymbol{x}}$ by solving (8).
4:     Compute $\hat{\boldsymbol{k}}$ by solving (10).
5:     $\beta \leftarrow \beta/1.1$
6: Output: Estimated blur kernel $\hat{\boldsymbol{k}}$ and skeleton image $\hat{\boldsymbol{x}}$.
```

where

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{x} * \hat{\boldsymbol{k}} - \boldsymbol{b}\|_2^2 + \beta\|\boldsymbol{x}\|_{RGTV} \quad (8)$$

or

$$\left(\hat{\boldsymbol{K}}^T\hat{\boldsymbol{K}} + 2\beta \cdot \boldsymbol{L}_\Gamma\right)\hat{\boldsymbol{x}} = \hat{\boldsymbol{K}}^T\boldsymbol{b} \quad (9)$$

where $\hat{\boldsymbol{K}}^T$ is the matrix representation of convolving with $\hat{\boldsymbol{k}}$. This is solved by the following algorithm:

Algorithm 6 Skeleton Image Restoration

```
1: Input: Blurry image $\boldsymbol{b}$ and estimated kernel $\hat{\boldsymbol{k}}$.
2: Initialize $\boldsymbol{L}_\Gamma$ as an unweighted graph Laplacian.
3: while not converge do
4:     Update $\hat{\boldsymbol{x}}$ using Conjugate Gradient method in (9).
5:     Update $\boldsymbol{L}_\Gamma = \boldsymbol{L}_\Gamma(\hat{\boldsymbol{x}})$.
6: Output: Restored skeleton image $\hat{\boldsymbol{x}}$.
```

where $\boldsymbol{L}_\Gamma$ denotes the Laplacian matrix computed from adjacency matrix Γ.
And to compute $\hat{\boldsymbol{k}}$ given $\hat{\boldsymbol{x}}$ solve:

$$\hat{\boldsymbol{k}} = \arg\min_{\boldsymbol{k}} \frac{1}{2}\|\nabla\hat{\boldsymbol{x}} * \boldsymbol{k} - \nabla\boldsymbol{b}\|_2^2 + \mu\|\boldsymbol{k}\|_2^2 \quad (10)$$

which is a quadratic convex function and has a closed form solution.

III. Experiments and Discussions

A. Image Deblurring

We compared the mentioned Blind Deconvolution algorithms using two images and conducted this experiment in two parts:

Part-1: Using a large 256×256 complex image: We use a 256×256 shape image with 23×23 a motion blur kernel to generate a blurry image. We will compare the performance of *CBD*, *PRIDA* and *GBD* on this blurry image. For *CBD*, we estimate the subspace of the original image from the Harr wavelet expansion of the blurry image and for the subspace of the blur kernel, we assume that we know its support and use corresponding columns of identity for it.
For *PRIDA*, we assume that we know the true size of the blur kernel i.e. 23×23 and set the parameter lambda to 10^{-6} for a sharper image. We run the algorithm for 1500 iterations.
For *GBD*, again we assume that we know the true size of the blur kernel i.e. 23×23. The algorithm runs for 4 levels, coarse to fine, 3 iterations at each level.
From Figure 1 we can see that all the algorithms provide surprisingly good results. We observe that *PRIDA* and *GBD* provided similar output, however *PRIDA* gave sharper image compared to *GBD*. We can see some overlapping image structure in Figure 1(c) and Figure 1(d), this is because of imperfect estimation of the blur kernel since the final image still has visible motion blur along the secondary diagonal direction.
CBD produces a noisy output as given in Figure 1(b). We can observe that Figure 1(b) has negligible motion blur but there is presence of significant noise. We believe that the noise came from our approximation of image subspace using only stronger Harr wavelet coefficients. Thus the resultant image presents the major attributes of the original image but lacks minor details.

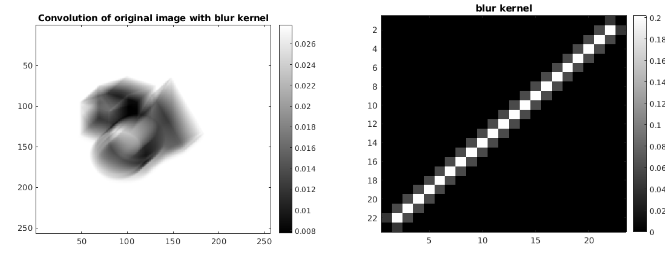
Part-2: Using a small 8×8 simple image: We use a 8×8 line image and blur it using a small 1×3 motion blur kernel for making a blurry image. We use this small image to evaluate the performance of *NCBD* and compare it with *CBD*. We are using a smaller image here because our implementation of *NCBD* requires a full image subspace matrix compared to a sparse matrix required for *CBD*. Since a normal image can have a large number of non-zero significant Harr wavelet coefficients, *NCBD* implementation suffers from memory unavailability for larger input images. We use the same method as in Part-1 to get the required subspace estimations for both *CBD* and *NCBD*. For this experiment we ran *NCBD* for 2000 iterations.

From Figure 2 we can see that both *CBD* and *NCBD* produced imperfect output, however one key difference is that while in *CBD* the imperfection is contained around the central object i.e. the line, in *NCBD* the noise is spread across the whole image. The blur kernel produced by both the algorithms is also almost similar, but *CBD* estimated a kernel which was closer to the original kernel.

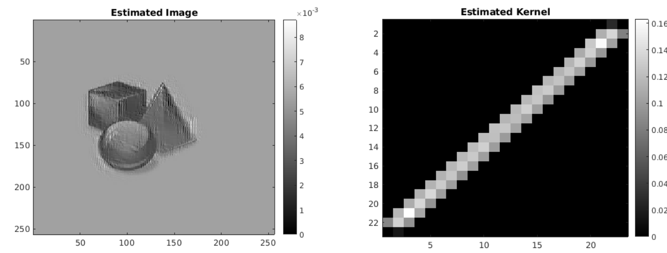
B. Phase transition

As in [1] and [2], we try to evaluate the boundary w.r.t the signal subspace sizes N and K for a effectiveness on the Blind Deconvolution algorithms using a fixed value of input signal length L. We use $L = 80$ and vary N and K from 2 to 40 using steps of size 2. We took the input vectors h and m randomly from a standard Gaussian vector with independent entries. We used a spare w vector and formed B by randomly choosing K columns from the $L \times L$ identity matrix. We used a dense $L \times N$ Gaussian random matrix for C.
We measure the effectiveness of the algorithms by plotting the (1- relative error) where relative error is measured as:

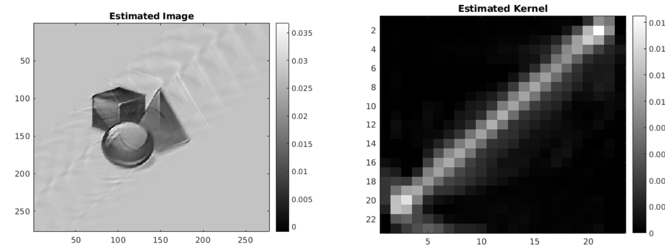
$$\text{Error} = \frac{\left\|\hat{\boldsymbol{X}} - \boldsymbol{w}\hat{\boldsymbol{x}}\right\|_F}{\|\boldsymbol{w}\hat{\boldsymbol{x}}\|_F}$$



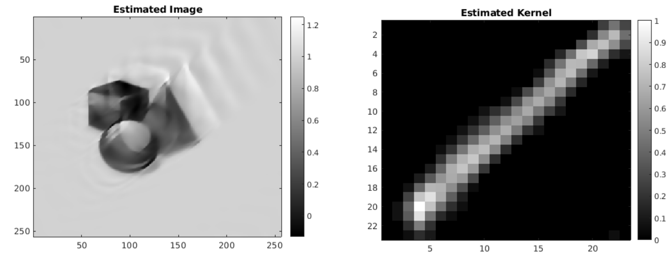
(a) Input Blurry image, True Blur Kernel



(b) Convex Blind Deconvolution

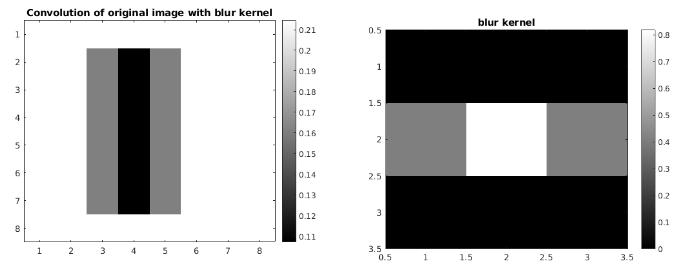


(c) PRIDA algorithm

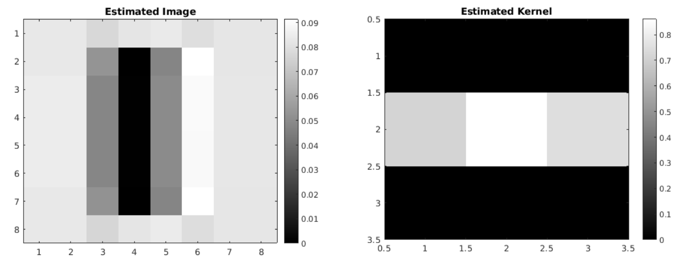


(d) Graph Blind Deconvolution

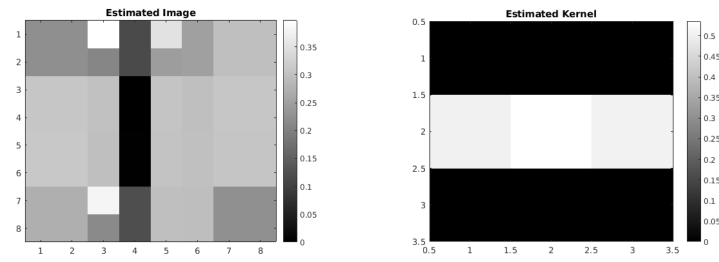
Fig. 1: Image deblurring comparision using larger image



(a) Input Blurry image, True Blur Kernel

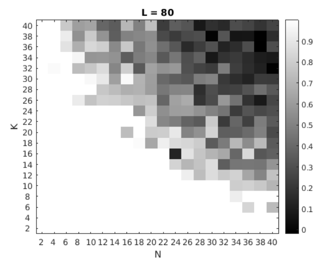


(b) Convex Blind Deconvolution

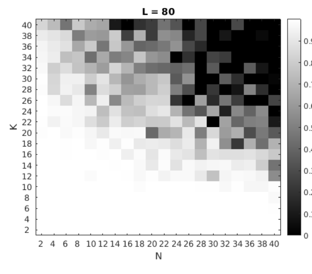


(c) Non-Convex Blind Deconvolution

Fig. 2: Image deblurring comparision using smaller image



(a) Convex Deconvolution



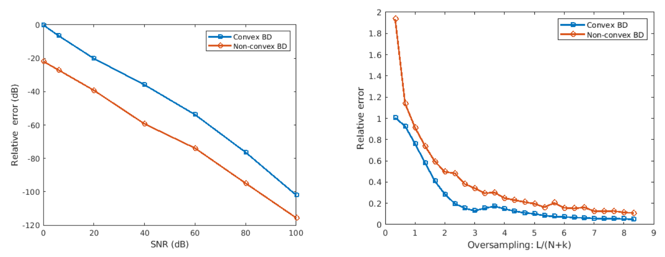
(b) Non-Convex Deconvolution

Fig. 3: Phase Transition

So in Figure 3, the closer the value to 1, the better is the deconvolution.
From the plots we can see that *CBD* provided consistently good results when $L \gtrsim 2.2(N+K)$ while *NCBD* performed well when $L \gtrsim 2.8(N+K)$.

C. Robustness comparison

We compare the performance of the convex and the non-convex algorithms in presence of random Gaussian noise. We compare the Signal to Noise Ratio (SNR) and the Oversampling Ratio with the relative error as done in [1]. For this



(a) Error vs SNR (b) Error vs Oversampling

Fig. 4: Robustness comparision

experiment, we use similar input setup of subspaces and signal vectors as we did in Experiment-2. For Part-1 we used $L = 2^8$, $N = 50$ and $k = 25$ and ran the experiment for 20 iterations to obtain average performance. On a log-log scale, we observe that for both *CBD* and *NCBD*, there is a linear relationship between the recovery error and the Signal to Noise Ratio. From Figure 4(a), we observe that *NCBD* performed better than *CBD* in presence of noise, indicating better robustness of the *NCBD* algorithm.
For Part-2 we used $N = 20$ and $K = 10$ and varied L from 10 to 250. As we did in the previous experiment, we again used 20 iterations to obtain average performance of both the algorithms.
We see from the Figure 4(b) that as the Oversampling ratio increases (by increasing L) both algorithms perform better and better. Here we see that *CBD* performed better than *NCBD* when comparing w.r.t to the Oversampling ratio, this result supports the conclusions we drew in the Phase transition comparison experiment.

IV. Conclusion

We studied four different Blind Deconvolution algorithms, each approaching the problem differently and making different input assumptions. We saw that although *CBD* performed better than *NCBD* in terms of relative error, *NCBD* was more robust and performed better in presence of noise. While in case of image deblurring, we observed *PRIDA* and *GBD* to perform better than *CBD* w.r.t. deblurred image but *CBD* recovered better blur kernel. After considering all the different comparisons we performed, we can say that there is no single best algorithm for the Blind Deconvolution problem and much more development can still be done in this field.

References

[1] A. Ahmed, B. Recht, and J. Romberg. Blind deconvolution using convex programming. Information Theory, IEEE Transactions on, 60(3):1711–1732, 2014.

[2] X. Li, S. Liang, T. Strohmer, and K. Wei, “Rapid robust and reliable blind deconvolution via nonconvex optimization,” Appl. Comput. Harmon. Anal., Feb. 2018.

[3] Ravi, S. N.; Mehta, R.; and Singh, V. 2018. Robust blind deconvolution via mirror descent. arXiv:1803.08137.

[4] Yuanchao Bai, Gene Cheung, Xianming Liu, and Wen Gao. Graph-based blind image deblurring from a single photograph. IEEE Trans. Image Process., 28(3):1404–1418, Oct. 2018.

V. Appendix

All the used plots and required code are provided in a zip file while submitting this term paper. Please have a look at the ReadMe file for code details and code bibliography.