Final Project of ELEC5470 FALL 2017 Optimization in Image Deblurring

Xiaoyun Yuan, 20322404

Abstract

In this report , I present some representative image deblurring algorithms and the convex/non-convex optimization techniques used in these papers.

Index Terms

Optimization, Image Deblurring, Prior

I. INTRODUCTION

Nowadays, smart phone has become the digital hub of everyone's daily life, and camera is one of the most important modules on smart phones. However, limited by the space of smart phones, the smart phone camera optic system is often very simple and CMOS size is usually very small. That means the image quality is easily affected environment, for example, low light condition leads to severe noise, camera shaking and long exposure time causes uncomfortable blur effect. In this report, I focus on how to remove the image blur effects caused by camera shaking.

Image deblurring is a fundamental computer vision research problem. Now, there are two different ways to solve the image deblurring problem: 1) traditional convolutional model approach; 2) neural network approach. In this report, I mainly focus on the first approach and give some brief introductions of the second approach.

A. Traditional Convolutional Model Approach

In traditional convolutional model, the image blur effect is expressed by $2D$ convolution:

$$\mathbf{b} = \mathbf{l} \otimes \mathbf{k} + \mathbf{n}, \tag{1}$$

where $\mathbf{b}$, $\mathbf{l}$ and $\mathbf{k}$ are the blurry image, latent sharp/unblurred image and blur kernel respectively. $\mathbf{n}$ denotes the noise. The blur kernel $\mathbf{k}$ is usually expressed by a small matrix which is related to the camera shaking trajectory, as shown in the Fig. 1. Because the 2D convolution is difficult to handle, Eqn. 1 is often written in frequency domain:

$$\mathcal{F}(\mathbf{b}) = \mathcal{F}(\mathbf{l}) \cdot \mathcal{F}(\mathbf{k}) + \mathcal{F}(\mathbf{n}), \tag{2}$$

where $\mathcal{F}(\cdot)$ denotes the Fourier transform, and $\cdot$ denotes element-wise product.
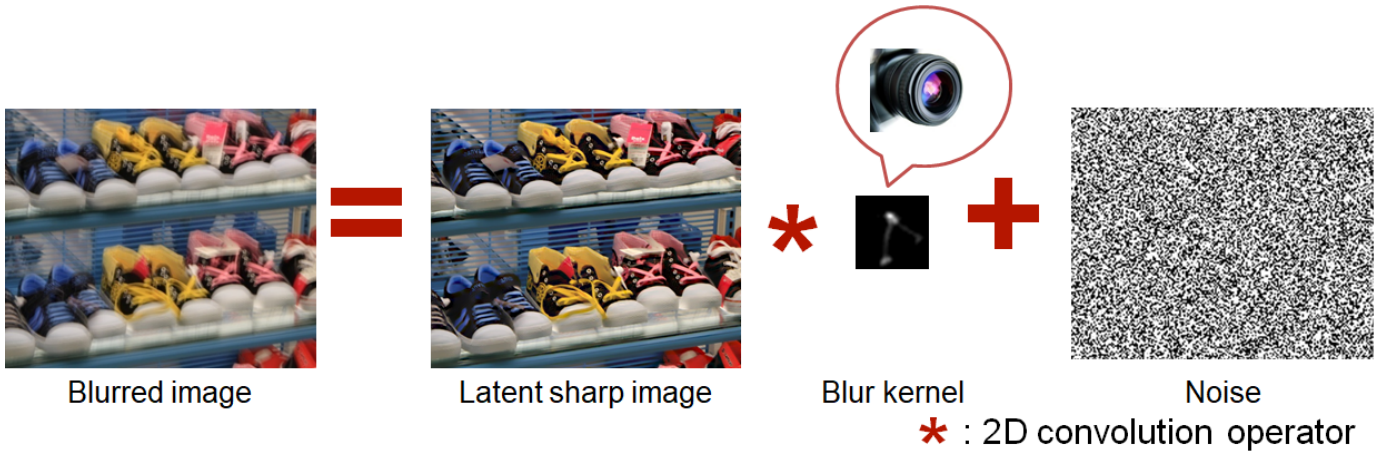

Fig. 1: Image Convolutional Blur Model

The deblurring problem can be divide into two kinds: 1) non-blind deblurring: only $\mathbf{l}$ is unknown, use existing $\mathbf{k}$ to solve latent sharp image $\mathbf{l}$:

$$\min_{\mathbf{l}} \|\mathbf{b} - \mathbf{l} \otimes \mathbf{k}\|. \tag{3}$$

2) Blind deblurring: both $\mathbf{l}$ and $\mathbf{k}$ are unknown:

$$\min_{\mathbf{l},\mathbf{k}} \|\mathbf{b} - \mathbf{l} \otimes \mathbf{k}\|, \tag{4}$$

The non-blind deblurring looks easy, but in fact, it is very tough in image deblurring problems. I recommend the readers to read the simple introduction [1]https://blogs.mathworks.com/steve/2007/08/13/image-deblurring-introduction/ first. The results are shown in Fig. 2: (a) and (d) show the blurry image and groundtruth image respectively. (b) is the deblurring result using pseudo inverse filter. We can see that although the main structure of the image is recovered, a lot of artifacts (ringing and noise) are introduced. (c) is the result of the state-of-the-art non blind deblurring algorithm [2], which I will introduce in next section.
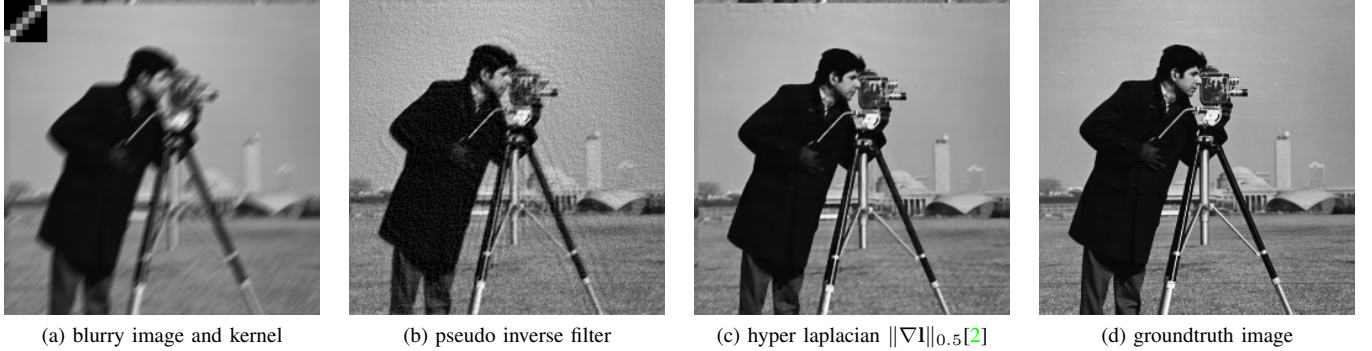

(a) blurry image and kernel   (b) pseudo inverse filter   (c) hyper laplacian $\|\nabla \mathbf{l}\|_{0.5}$[2]   (d) groundtruth image

Fig. 2: Image Deblurring Example

To solve the problem is Fig. 2(b), existing state-of-the-art algorithms usually solve the following optimization problem with image prior ϕ and kernel prior ρ instead of original one: 1) Non-blind deblurring

$$\min_{\mathbf{l}} \|\mathbf{b} - \mathbf{l} \otimes \mathbf{k}\| + \lambda\phi(\mathbf{l}). \tag{5}$$

2) Blind deblurring

$$\min_{\mathbf{l},\mathbf{k}} \|\mathbf{b} - \mathbf{l} \otimes \mathbf{k}\| + \lambda\phi(\mathbf{l}) + \mu\rho(\mathbf{k}). \tag{6}$$

In blind deblurring, Eqn. 6 will be split into two minimization problems and solve $\mathbf{l}$ and $\mathbf{k}$ alternately:

$$\begin{cases} \min_{\mathbf{l}} & \|\mathbf{b} - \mathbf{l} \otimes \mathbf{k}\| + \lambda\phi(\mathbf{l}) \\ \min_{\mathbf{k}} & \|\mathbf{b} - \mathbf{l} \otimes \mathbf{k}\| + \mu\rho(\mathbf{k}) \end{cases} \tag{7}$$

An important thing to notice: although there are two optimization variables l, k in the optimization problem, the purpose is to get a good k only. After solving this problem, use k to solve non-blind deblurring problem to get the final sharp image. Note that the prior $\phi \mathbf{l}$ used in blind and non-blind deblurring are different. I will give more details in section. II.

Nowadays, the image are very large. For example, a smart phone with 12MP could capture 3968×2976 images, which means usually very large kernels are needed to estimate (larger than 120×120). To directly estimate a variable with so high dimension is very difficult, so most blind deblurring algorithms choose a multi-scale coarse-to-fine approach: estimating a smalling kernel using low resolution image first and resizing to high resolution as the initial value.

B. Neural Network Approach

The convolutional model looks very beautiful, but solving it is really not easy. Therefore, some researchers proposed to use neural network to predict the latent sharp image directly. For example, the neural network in Fig. 3 takes the stacked nearby videos frames as input, and directly output the deblurred central video.

Until now, the neural network is still a black box to researchers. No one could give a theoretical explanation, so I am not going to talk the details here.

II. OVERVIEW OF EXISTING WORK

In this section, I categorize the deblurring algorithms into 3 kinds: 1) image priors; 2) kernel priors; 3) neural networks priors, and briefly introduce some representative works in this section.
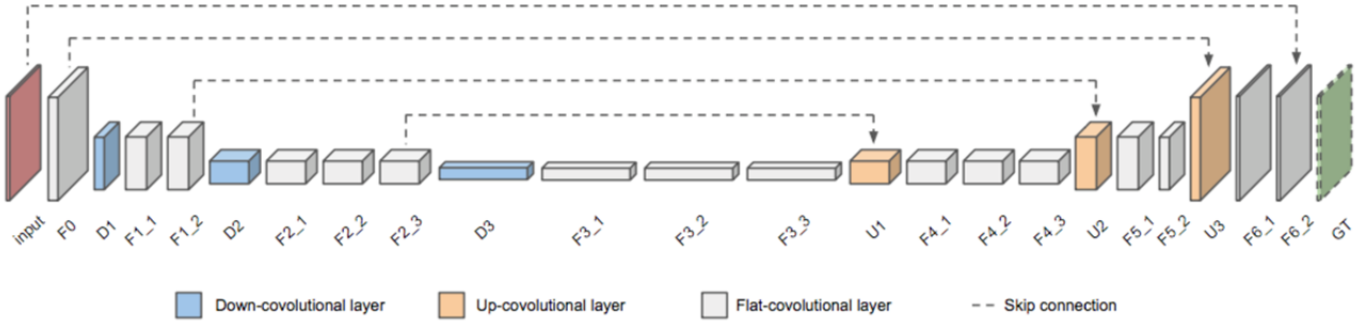
Fig. 3: Structure of CNN used in Deep Video Deblurring[3]

A. Image Priors

As mentioned above, I formulate the image deblurring objective function as following minimization problem:

$$\min_{\mathbf{l},\mathbf{k}} \|\mathbf{b} - \mathbf{l} \otimes \mathbf{k}\| + \lambda\phi(\mathbf{l}) + \mu\rho(\mathbf{k}). \tag{8}$$

Hyper-laplacian Prior[2]: this paper proposed a excellent non-blind image deblurring algorithm. Kernel is already given in non-blind deblurring so only need to solve the first problem in Eqn. 7. A hyper-laplacian image prior $\phi(\mathbf{l}) = \|\nabla\mathbf{l}\|_\alpha$ is proposed in this paper. As shown in Fig. 4, the dark blue curve denotes the empirical image gradient distribution of natural images. The reason why hyper-laplacian is proposed is that the curve shape of hyper-laplacian (green) is closet to the empirical one (dark blue).
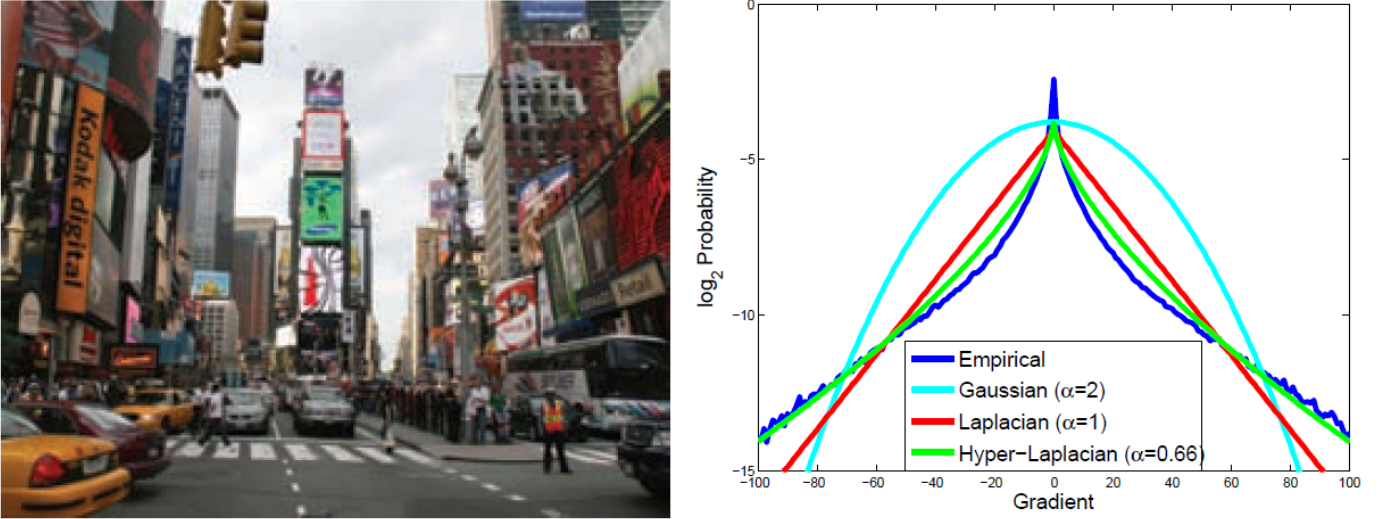

Fig. 4: Hyper-Laplacian Image Prior[3]

However, solving the problem is still not trivial because the image prior $\phi(\mathbf{l}) = \|\nabla\mathbf{l}\|_\alpha$ is non-convex. The most commonly used method in image deblurring is to re-formulate the objective function again:

$$\min_{\mathbf{l}} \|\mathbf{b} - \mathbf{l} \otimes \mathbf{k}\|_2^2 + \lambda_1\|\nabla\mathbf{l} - \mathbf{g}\|_2^2 + \lambda_2\|\mathbf{g}\|_\alpha. \tag{9}$$

Eqn. 9 can be divided into two sub-optimization problems and solve alternately:

$$\begin{cases} \min_{\mathbf{l}} & \|\mathbf{b} - \mathbf{l} \otimes \mathbf{k}\|_2^2 + \lambda_1\|\nabla\mathbf{l} - \mathbf{g}\|_2^2 \\ \min_{\mathbf{g}} & \lambda_1\|\nabla\mathbf{l} - \mathbf{g}\|_2^2 + \lambda_2\|\mathbf{g}\|_\alpha. \end{cases} \tag{10}$$

The first minimization problem if Eqn. 10 is a least-square problem which has closed-form solution:

$$\mathbf{l} = \mathcal{F}^{-1}\left(\frac{\overline{\mathcal{F}(\mathbf{k})}\mathcal{F}(\mathbf{b}) + \lambda_1 F_G}{\overline{\mathcal{F}(\mathbf{k})}\mathcal{F}(\mathbf{k}) + \lambda_1\overline{\mathcal{F}(\nabla)}\mathcal{F}(\nabla)}\right), \tag{11}$$

where $F_G = \overline{\mathcal{F}(\nabla_h)}\mathcal{F}(\mathbf{g}_h) + \overline{\mathcal{F}(\nabla_v)}\mathcal{F}(\mathbf{g}_v)$. ∇_h and ∇_v represent the horizontal and vertical differential operators. The second problem is difficult, exact analytical solutions can only be found for some specific value, e.g. $\alpha = 0.5, 0.66$. For general α values, this paper proposed a lookup table solution.

Unnatural L_0 Sparsity Prior[4]: this paper proposed a blind image blurring algorithm with objective function as follows:

$$\min_{\mathbf{l},\mathbf{k}} \|\mathbf{b} - \mathbf{l} \otimes \mathbf{k}\|_2^2 + \lambda\|\nabla\mathbf{l}\|_0 + \mu\|\mathbf{k}\|_2^2. \tag{12}$$

Similar as the hyper-laplacian prior paper, this large optimization problem can be re-formulated into 3 small sub-problems and solved alternately. The kernel is estimated in gradient space $\nabla\mathbf{l}$, $\nabla\mathbf{b}$ to replace $\mathbf{l}$, $\mathbf{b}$, because estimating kernel in gradient space is more accurate[5].

$$\begin{cases} \min_{\mathbf{l}} & \|\mathbf{b} - \mathbf{l} \otimes \mathbf{k}\|_2^2 + \lambda_1\|\nabla\mathbf{l} - \mathbf{g}\|_2^2 \\ \min_{\mathbf{g}} & \lambda_1\|\nabla\mathbf{l} - \mathbf{g}\|_2^2 + \lambda_2\|\mathbf{g}\|_0 \\ \min_{\mathbf{k}} & \|\nabla\mathbf{b} - \nabla\mathbf{l} \otimes \mathbf{k}\|_2^2 + \mu\|k\|_2^2. \end{cases} \tag{13}$$

The first sub-problem is same as the Eqn. 10. The third sub-problem also has closed form solution:

$$\mathbf{k} = \mathcal{F}^{-1}\left(\frac{\overline{\mathcal{F}(\nabla_h\mathbf{l})}\mathcal{F}(\nabla_h\mathbf{b}) + \overline{\mathcal{F}(\nabla_v\mathbf{l})}\mathcal{F}(\nabla_v\mathbf{b})}{\overline{\mathcal{F}(\nabla_h\mathbf{l})}\mathcal{F}(\nabla_h\mathbf{l}) + \overline{\mathcal{F}(\nabla_v\mathbf{l})}\mathcal{F}(\nabla_v\mathbf{l}) + \mu} \right). \tag{14}$$

Sometimes, the close form solution is not stable, conjugate gradient method can be used to estimate a more stable result.

The key contribution of this paper is to solve the second sub-problem of Eqn. 13. The second sub-problem looks difficult but it is an element-wise minimization problem and very easy to solve. For every element $\mathbf{g}_i$ in $\mathbf{g}$, only two values can be chosen: 0 or $\nabla\mathbf{l}_i$. If choosing 0, cost function value equals $\lambda_1|\nabla\mathbf{l}_i|^2$. If choosing $\nabla\mathbf{l}_i$, cost function value equals λ_2. So, the second sub-problem can be solve by:

$$\mathbf{g}_i = \begin{cases} \nabla\mathbf{l}_i, & |\nabla\mathbf{l}_i|^2 \geqslant \frac{\lambda_2}{\lambda_1}, \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

L_0 ***Regularized Intensity and Gradient Prior***[6]: an extension of unnatural L_0 sparsity prior[4] used for text image deblurring: $\phi(\mathbf{l}) = \alpha\|\mathbf{l}\|_0 + \|\nabla\mathbf{l}\|$. The solving method is nearly the same as that in [4], only needs more iterations to refine $\mathbf{l}$ and $\nabla\mathbf{l}$ alternately.

Normalized Sparsity Prior L_1/L_2 [7]: this paper proposed a image prior called normalized sparsity prior $\phi(\mathbf{l}) = \frac{\|\nabla\mathbf{l}\|_1}{\|\nabla\mathbf{l}\|_2}$, and the objective function becomes:

$$\min_{\mathbf{l},\mathbf{k}} \|\mathbf{b} - \mathbf{l} \otimes \mathbf{k}\|_2^2 + \lambda\frac{\|\nabla\mathbf{l}\|_1}{\|\nabla\mathbf{l}\|_2} + \mu\|\mathbf{k}\|_1. \tag{16}$$

The reason why use normalize sparsity as image prior is that L_1/L_2 is a quite good measure of image sharpness, as shown in Fig. 5.
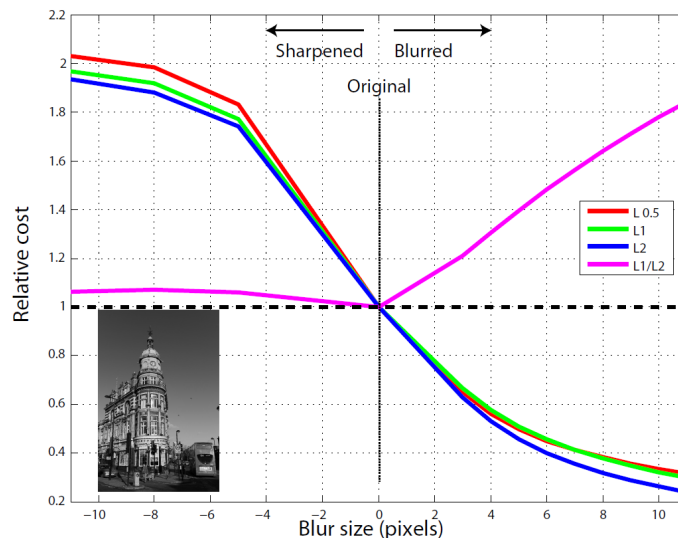

Fig. 5: Normalized Sparsity Measure $\frac{L_1}{L_2}$

Similarly, Eqn. 16 can be divide into the following two sub-problems:

$$\begin{cases} \min_{\mathbf{l}} & \|\mathbf{b} - \mathbf{l} \otimes \mathbf{k}\|_2^2 + \lambda_1 \frac{\|\nabla \mathbf{l}\|_1}{\|\nabla \mathbf{l}\|_2} \\ \min_{\mathbf{k}} & \|\nabla \mathbf{b} - \nabla \mathbf{l} \otimes \mathbf{k}\|_2^2 + \mu \|k\|_1. \end{cases} \tag{17}$$

Different from the above mentioned papers, L_1 norm is used here to estimate kernel. No close form solution but it can be solved by iterative re-weighted least squares (IRLS) easily. The first sub-problem is non-convex and very difficult to solve directly. To make it solvable, this paper design an iterative optimization algorithm with inner and outer loops: in inner loop, first fix the denominator $\|\nabla \mathbf{l}\|_2$ and this problem becomes a convex L_1 regularized problem which can be solved by iterative shrinkage-thresholding algorithm (ISTA)[8]. In outer loop, re-estimate the denominator $\|\nabla \mathbf{l}\|_2$.

B. Kernel Priors

All the papers in Section. II-A use very simple kernel prior ($\|k\|_2^2$ or $\|k\|_1$), which are convex and very easy to solve. However, it is not good enough sometimes, as illustrated in Fig. 6. Small noise or fake branches in kernel can cause severe visual artifacts. Therefore, some paper propose specific kernel prior, for example, using IMU (inertial measurement sensor) to
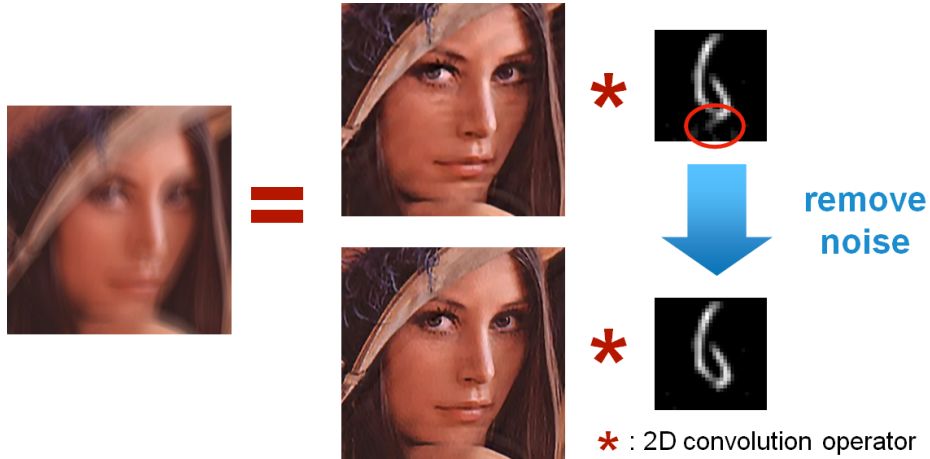

Fig. 6: Effect of Kernel Noise

help estimate the kernel shape. In this report, we introduce a simple optimization based approach:

Separable Kernel Prior[9]: this paper proposed to force the kernel in a curve region by extract the trajectory of the kernel, and the original kernel optimization sub-problem becomes

$$\begin{cases} \min_{\mathbf{K}} & \|\nabla \mathbf{b} - \nabla \mathbf{l} \otimes \mathbf{k}\|_2^2 + \mu \|W \odot \mathbf{k}\|_1 \\ \text{s.t.} & W = 1 - E(T). \end{cases} \tag{18}$$
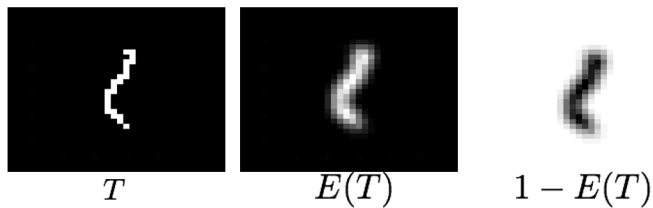

Fig. 7: Refine Kernel Estimation using Trajectory

As presented in Fig. 7, where *odot* is element-wise product, T denotes the trajectory of the original kernel, $E(\cdot)$ represent the Gaussian mask and W is the mask used to remove noise and fake branches in kernel.

C. Neural Networks Priors

The image priors proposed in Section. II-A are very simple statistics, which can be solved by existing optimization techniques but have many failure cases unavoidably. Thus, some researchers proposed to use neural network as image prior. In this report, I introduce a non-iterative neural approach.

Non-iterative Neural Approach[10]: the methods in Section. II-A derive latent sharp image l update rule from objective functions and the image priors are used as an image sharpness measure. However, it is infeasible for neural network priors because of its high-complexity. Therefore, this paper proposed to train a neural network to update the latent sharp image l directly, as shown in Fig. 8. The neural network takes blurred image patches as input and output the filter coefficients of kernel directly. After estimating the kernel, the final deblurred result can be obtained by state-of-the-art non-blind deblurring algorithm. This method is non-iterative and highly parallable, which is friendly to GPU hardware.
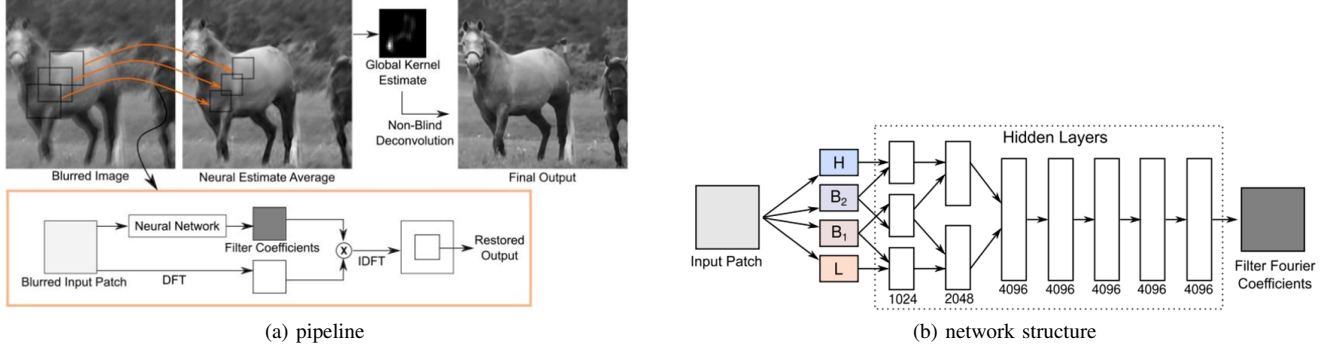

(a) pipeline   (b) network structure

Fig. 8: Non-iterative Neural Approach Pipeline and Network Structure

D. Summary

I have introduced some representative image deblurring works in this section, and Lai *et al.* published a paper which gave a comparative study for single image blind deblurring[11]. In that paper, they systematically analyzed the performance of 13 representative blind deblurring algorithms using both subjective (via user study) and objective quality measures. Finally, they presented the following observations:

- **Image priors**: Sparse gradient priors [4] and intensity [6] are more reliable and effective than other image priors.
- **Blur models**: The widely used convolutional blur models can not deal with saturated regions and non-Gaussian noise well.
- **Datasets**: Existing methods achieve very good performance on synthetic images. But real blurry images are still difficult to handle.

In this report, I will focus on sparse gradient priors, give more details of the optimization procedures and try to make some improvement.

III. CRITICISM OF THE EXISTING WORK

Before introducing my work, I give some drawbacks of existing algorithms from the optimization perspective (not image processing perspective) as this is a report of convex optimization course.

A. No Convergence Criterion

Usually blind deblurring algorithms have no clear convergence criterion. I think the main reason is that it is extremely difficult to give a accurate quality score of the deblurred image or blur kernel. Therefore, many algorithms just fix the iteration numbers, as shown in Fig. 9. The left column and middle column are L_0 sparse prior, which convergence very fast. Usually only 5 iterations are need. However, the drawback is obviously, too. Because of lacking stop criterion, there is no guarantee that the result is global/local optimal.

B. Different Objective Functions in Estimating Kernel and Final Sharp Image

As I mentioned in Section. I (red words), blind deblurring can be divided two steps: 1) solve Eqn. 7 to get a good kernel. 2) use the solved kernel to get final sharp image by solving Eqn. 5. The image prior $\phi(l)$ used in Eqn. 5 and Eqn. 7 are different, which looks weird.

For example, the L_0 deblurring paper, $\phi(l) = \|\nabla l\|_0$ in blind step and $\phi(l) = \|\nabla l\|_{0.5}$ in non-blind step. The intermediate L_0 regularized sharp image is shown Fig. 10(b). The gradient map of an natural image should not be L_0 sparse, however, it is a very effective prior to help estimate kernel.
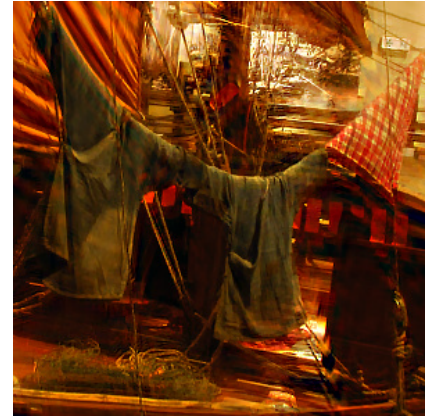
IV. NEW CONTRIBUTION

In this section, I will present some new contributions of my report.

Algorithm 1: L_0 Deblurring in One Image Level

input : blurred image y

output: blur kernel k, deblurred image $\tilde{x}$

1   initialize k^1 from the coarser-scale kernel estimate

2   **for** $t = 1 : 5$ **do**

3   // update image

4     $\epsilon \leftarrow 1$

5   **for** $i = 1 : 4$ **do**

6   **for** $j = 1 : \epsilon^{-1}$ **do**

7   solve for l using Eq. (11)

8   solve for $\tilde{x}^{t+1}$ using Eqs. (12) or (14)

9   **end**

10      $\epsilon \leftarrow \epsilon/2$ // graduate non-convexity

11   **end**

12   // update kernel

13   solve for k^{t+1} using Eqs. (15) or (17)

14 **end**

Algorithm 1 Solving (6)

Input: Blur image y and blur kernel k.

$x \leftarrow y$, $\beta \leftarrow 2\lambda\sigma$.

repeat

   solve for u using (10).

   $\mu \leftarrow 2\lambda$.

   repeat

     solve for g using (11).

     solve for x using (8).

     $\mu \leftarrow 2\mu$.

   until $\mu > \mu_{\max}$

   $\beta \leftarrow 2\beta$.

until $\beta > \beta_{\max}$

Output: Intermediate latent image x.

Algorithm 2 Blur kernel estimation algorithm

Input: Blur image y.

initialize k with the results from the coarser level.

for $i = 1 \to 5$ **do**

   solve for x using Algorithm 1.

   solve for k using (12).

   $\lambda \leftarrow \max\{\lambda/1.1, 1e^{-4}\}$.

end for

Output: Blur kernel k and intermediate latent image x.

Algorithm 2 : Iterative Shrinkage-Thresholding Algorithm (ISTA)

Require: Operator K, regularization parameter λ

Require: Initial iterate x^0, observed image y

Require: Threshold t, maximum iterations N

1:   **for** $j = 0$ to $N - 1$ **do**

2:     $v = y - tK^T(Kx^j - y)$

3:     $x^{j+1} = S_{t\lambda}(v)$

4:   **end for**

5:   **return** Output image x^N

Algorithm 3 : x Update

Require: Blur kernel k from previous k update

Require: Image x^0 from previous x update

Require: Regularization parameter $\lambda = 20$

Require: Maximum outer iterations $M = 2$, inner its. $N = 2$

Require: ISTA threshold $t = 0.001$

1:   **for** $j = 0$ to $M - 1$ **do**

2:     $\lambda' = \lambda \|x^j\|_2$

3:     $x^{j+1} = \text{ISTA}(k, \lambda', x^j, t, N)$

4:   **end for**

5:   **return** Updated image x^M.

Fig. 9: Optimization Algorithms Details

(a) blurry image   (b) intermediate L_0 regularized sharp image   (c) final deblurred image

Fig. 10: Intermediate Result of L_0 Deblurring

A. New Implementation of L_0 Sparse Solver

We have learned how to solve L_0 sparse optimization this semester. In class, the original problem is re-written to a L_1 weighted optimized problem and solved by iterative re-weighted L_1-norm heuristic solver. But in these image deblurring paper, a different solver is used which looks more specific, easier and faster.

However, to train my skills, I decide to do two small experiments first[1]:

- Implement a new solver using L_1 sparse image prior and compare the result with original L_0 solver.
- Implement a L_0 solver using the method in class: iterative re-weighted L_1-norm heuristic method, and compare with original L_0 solver.

1) Implement L_1 solver: Inspired by Eqn. 13, we split the original L_1 regularized optimization problem in two sub-problems:

$$\begin{cases} \min_{\mathbf{l}} & \|\mathbf{b} - \mathbf{l} \otimes \mathbf{k}\|_2^2 + \lambda_1 \|\nabla \mathbf{l} - \mathbf{g}\|_2^2 \\ \min_{\mathbf{g}} & \lambda_1 \|\nabla \mathbf{l} - \mathbf{g}\|_2^2 + \lambda_2 \|\mathbf{g}\|_1 \end{cases} \tag{19}$$

Same as L_0 sparse image prior [4][6], the second sub-problem with L_1 image prior is also an element-wise optimization problem. Which can be solved quickly by:

$$\mathbf{g}_i = \begin{cases} \nabla \mathbf{l}_i, & |\nabla \mathbf{l}_i|^2 \geqslant \frac{\lambda_2}{\lambda_1}, \\ \nabla \mathbf{l}_i, & |\nabla \mathbf{l}_i|^2 \geqslant \frac{\lambda_2}{\lambda_1}, \\ 0, & \text{otherwise.} \end{cases} \tag{20}$$

---

[1]Here, I want to thank Prof. PAN, Jinshan (https://sites.google.com/site/jspanhomepage/) who provides many useful codes which helps me a lot in finishing this project.

For kernel estimation step, we use L_2 kernel prior which has close-form solution Eqn. 14.

2) Implement L_0 solver using Iterative Re-weighted L_1-norm Heuristic Method: In lecture 13 of this semester, we have learned that the L_0 optimization problem can be solved by **iterative re-weighted L_1-norm heuristic method**. Therefore, I want to use this algorithm to implement another solver and compare with the original one.

Based on the algorithm, we only need to multiple a weight on the L_1 term and solve the L_1 regularized optimization problem. The second sub-problem becomes:

$$\min_{\mathbf{g}} \quad \lambda_1 \|\nabla \mathbf{l} - \mathbf{g}\|_2^2 + \lambda_2 \sum_i |\omega_i \mathbf{g_i}| \tag{21}$$

$$\text{subject to} \quad \omega_i = 1/(\epsilon + |\nabla \mathbf{l}_i|).$$

Similarly, this problem can be solved easily:

$$\mathbf{g}_i = \begin{cases} \nabla \mathbf{l}_i, & |\nabla \mathbf{l}_i|^2 \geqslant \frac{\lambda_2}{\lambda_1}, \\ \nabla \mathbf{l}_i, & |\nabla \mathbf{l}_i|^2 \geqslant \frac{\lambda_2}{\lambda_1}, \\ 0, & \text{otherwise}. \end{cases} \tag{22}$$

B. One Step Further: Deblurring with Maximum Saturation Feature

V. NUMERICAL RESULTS

VI. CONCLUSION

REFERENCES

[1] S. Eddins, "Image deblurring introduction," *https://blogs.mathworks.com/steve/2007/08/13/image-deblurring-introduction/*.

[2] D. Krishnan and R. Fergus, "Fast image deconvolution using hyper-laplacian priors," in *Advances in Neural Information Processing Systems*, 2009, pp. 1033–1041.

[3] S. Su, M. Delbracio, J. Wang, G. Sapiro, W. Heidrich, and O. Wang, "Deep video deblurring," *arXiv preprint arXiv:1611.08387*, 2016.

[4] L. Xu, S. Zheng, and J. Jia, "Unnatural l0 sparse representation for natural image deblurring," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 1107–1114.

[5] S. Cho and S. Lee, "Fast motion deblurring," in *ACM Transactions on Graphics (TOG)*, vol. 28, no. 5. ACM, 2009, p. 145.

[6] J. Pan, Z. Hu, Z. Su, and M.-H. Yang, "Deblurring text images via l0-regularized intensity and gradient prior," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2901–2908.

[7] D. Krishnan, T. Tay, and R. Fergus, "Blind deconvolution using a normalized sparsity measure," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 233–240.

[8] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[9] L. Fang, H. Liu, F. Wu, X. Sun, and H. Li, "Separable kernel for image deblurring," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2885–2892.

[10] A. Chakrabarti, "A neural approach to blind motion deblurring," in *European Conference on Computer Vision*. Springer, 2016, pp. 221–235.

[11] W.-S. Lai, J.-B. Huang, Z. Hu, N. Ahuja, and M.-H. Yang, "A comparative study for single image blind deblurring," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1701–1709.