Blind Deconvolution via Total Variation Regularization

Komal Krishna Mogilipalepu

I. Problem Setup

Consider a measurement model

$$\mathbf{B} = \mathbf{K} * \mathbf{S} + \mathbf{N}, \tag{1}$$

where $\mathbf{S} : hs \times ws$ is a sharp image, $\mathbf{K} : hk \times wk$ is a blur kernel, and $\mathbf{N}$ is a i.i.d. Gaussian noise. $\mathbf{K} * \mathbf{S}$ denotes convolution without padding between $\mathbf{K}$ and $\mathbf{S}$, and $\mathbf{B} : (hs - hk + 1) \times (ws - wk + 1)$ is the blurred image. The blind deconvolution problem is to recover the blur kernel and underlying sharp image from its blurred one. The simplest approach to solve this problem is to maximize the posterior distribution

$$\begin{aligned}(\widehat{\mathbf{S}}, \widehat{\mathbf{K}}) &= \arg\max_{\mathbf{S},\mathbf{K}} p(\mathbf{S}, \mathbf{K}|\mathbf{B}) \\ &= \arg\max_{\mathbf{S},\mathbf{K}} p(\mathbf{B}|\mathbf{S}, \mathbf{K}) p(\mathbf{S}) p(\mathbf{K}),\end{aligned} \tag{2}$$

where $p(\mathbf{B}|\mathbf{S}, \mathbf{K})$ models the distribution of noisy blurry image given $(\mathbf{S}, \mathbf{K})$, $p(\mathbf{S})$ models the distribution of sharp image, and $p(\mathbf{K})$ is the prior knowledge about the blur kernel. The problem (2) under the assumption of Gaussian noise can be written as

$$\begin{aligned}(\widehat{\mathbf{S}}, \widehat{\mathbf{K}}) = \arg\min_{\mathbf{S},\mathbf{K}} \quad & \|[\mathbf{K} * \mathbf{S}](\mathbf{x}) - \mathbf{B}(\mathbf{x})\|^2_{L_2(\mathbf{x})} \\ & + \lambda \mathcal{R}(\mathbf{S}(\mathbf{x})) + \gamma \mathcal{J}(\mathbf{K}(\mathbf{x})),\end{aligned} \tag{3}$$

where $\mathbf{x}$ is the support of the corresponding matrix, the data fit term corresponds to the log-likelihood $\log(p(\mathbf{B}|\mathbf{S}, \mathbf{K}))$, the regularizations $\mathcal{R}(\mathbf{S}(\mathbf{x}))$ and $\mathcal{J}(\mathbf{K}(\mathbf{x}))$ are the smoothness priors for $\mathbf{S}$ and $\mathbf{K}$. λ and γ are the positive parameters that measure the amount of regularization. Additionally, the following constraints on $\mathbf{K}(\mathbf{x})$ improve the convergence of the algorithm

$$\mathbf{K}(\mathbf{x}) \succcurlyeq 0, \quad \|\mathbf{K}(\mathbf{x})\|_{L_1(\mathbf{x})} = 1 \tag{4}$$

Therefore, considering the constraints in (4) and without any smoothness constraints on blur kernel, we study the following optimization problem

$$\begin{aligned}\min_{\mathbf{S},\mathbf{K}} \quad & \|[\mathbf{K} * \mathbf{S}](\mathbf{x}) - \mathbf{B}(\mathbf{x})\|^2_{L_2(\mathbf{x})} + \lambda \mathcal{R}(\mathbf{S}(\mathbf{x})) \\ \text{s.t} \quad & \mathbf{K}(\mathbf{x}) \succcurlyeq 0, \quad \|\mathbf{K}(\mathbf{x})\|_{L_1(\mathbf{x})} = 1,\end{aligned} \tag{5}$$

where $\mathbf{x} = [x \; y]^T$, $\mathcal{R}(\mathbf{S}(\mathbf{x}))$ is the total variation regularization on $\mathbf{S}(\mathbf{x})$. Total variation regularization is the sum of the L_1 norm on the gradient components of $\mathbf{S}(\mathbf{x})$ or the sum of the L_2 norm on the gradient components of $\mathbf{S}(\mathbf{x})$. By this, minimizing the total variation regularization of an image enforces the recovered image to be smooth. In a discrete setting, $\mathcal{R}(\mathbf{S}(\mathbf{x})) = \|\nabla \mathbf{S}(\mathbf{x})\|_{1,1}$ or $\mathcal{R}(\mathbf{S}) = \|\nabla \mathbf{S}(\mathbf{x})\|_{2,1}$, $\nabla \mathbf{S}(\mathbf{x}) = [\nabla_x \mathbf{S}(\mathbf{x}) \; \nabla_y \mathbf{S}(\mathbf{x})]^T$ is the discrete gradient of $\mathbf{S}(\mathbf{x})$, $\nabla_x \mathbf{S}(\mathbf{x}) = \mathbf{S}(x+1, y) - \mathbf{S}(x, y)$, and

$$\|\nabla \mathbf{S}(\mathbf{x})\|_{p,q} = \big(\sum_{\mathbf{x}} \|\nabla \mathbf{S}(\mathbf{x})\|_p^q\big)^{\frac{1}{q}}. \tag{6}$$

In the above optimization problem (5), we consider $\mathcal{R}(\mathbf{S}(\mathbf{x})) = \|\nabla \mathbf{S}(\mathbf{x})\|_{2,1} = \sum_{\mathbf{x}} \|\nabla \mathbf{S}(\mathbf{x})\|_2$.

Algorithm 1: projected alternating minimization algorithm for blind deconvolution

Input: $\mathbf{B}$, hk, wk, λ_{init}, λ_{min}.
Output: $\widehat{\mathbf{S}}$, $\widehat{\mathbf{K}}$

```
Initialize Ŝ ← pad(B)
Initialize K̂ ← ones(hk,wk)/(hk * wk)
Initialize λ ← λ_init
while not converged do
    S^{t+1} = S^t − η_s ( K^t_− * (K^t * S^t − B) + λ∇ · ∇S^t / ‖∇S^t‖_2 )
    K^{t+1/3} = K^t − η_k ( S^{t+1}_− * (K^t * S^{t+1} − B) )
    K^{t+2/3} = max{K^{t+1/3}, 0}
    K^{t+1} = K^{t+2/3} / ‖K^{t+2/3}‖_1
    λ ← max{0.999 * λ_init, λ_min}
    t ← t + 1
end
Ŝ ← S^{t+1}
K̂ ← K^{t+1}
```

where the update steps are

$$\mathbf{S}^{t+1} = \mathbf{S}^t - \eta_s\Big(\mathbf{K}^t_- * \big(\mathbf{K}^t * \mathbf{S}^t - \mathbf{B}\big) + \lambda \nabla \cdot \frac{\nabla \mathbf{S}^t}{\|\nabla \mathbf{S}^t\|_2}\Big)$$

$$\mathbf{K}^{t+1/3} = \mathbf{K}^t - \eta_k\Big(\mathbf{S}^{t+1}_- * \big(\mathbf{K}^t * \mathbf{S}^{t+1} - \mathbf{B}\big)\Big)$$

$$\mathbf{K}^{t+2/3} = \max\{\mathbf{K}^{t+1/3}, 0\}, \quad \mathbf{K}^{t+1} = \frac{\mathbf{K}^{t+2/3}}{\|\mathbf{K}^{t+2/3}\|_1}$$

II. Method

The formulation in eq.(5) involves solving a constrained nonconvex optimization problem, and also, the solution space contains shifted versions of sharp image kernel pair as well, i.e., if $(\widehat{\mathbf{S}}, \widehat{\mathbf{K}})$ is a solution then $(\widehat{\mathbf{S}}(\mathbf{x} + \mathbf{c}), \widehat{\mathbf{K}}(\mathbf{x} + \mathbf{c}))$ for any $\mathbf{c} \in \mathbf{R}^2$ are also the solutions. Alternating minimization is a general approach to solving this optimization problem.

A. *Usual Alternating Minimization*

The usual alternating minimization alternates between estimating the sharp image given the kernel and estimating the kernel with the sharp image. The problem of recovering a sharp image given the kernel leads to an unconstrained convex optimization problem in $\mathbf{S}$

$$\mathbf{S}^{t+1}(\mathbf{x}) = \arg\min_{\mathbf{S}} \|[\mathbf{K}^t * \mathbf{S}](\mathbf{x}) - \mathbf{B}(\mathbf{x})\|^2_{L_2(\mathbf{x})} + \lambda \mathcal{R}(\mathbf{S}(\mathbf{x})) \tag{7}$$

and recovering a kernel given the sharp image leads to a constrained convex problem in $\mathbf{K}$

$$\mathbf{K}^{t+1}(\mathbf{x}) = \arg\min_{\mathbf{K}} \|[\mathbf{K} * \mathbf{S}^t](\mathbf{x}) - \mathbf{B}(\mathbf{x})\|^2_{L_2(\mathbf{x})}$$
$$\text{s.t} \quad \mathbf{K}(\mathbf{x}) \succcurlyeq 0, \quad \|\mathbf{K}(\mathbf{x})\|_{L_1(\mathbf{x})} = 1. \tag{8}$$

Unfortunately, the above alternating minimization could get stuck at a no-blur solution if we don't initialize carefully. To avoid this, a projected alternating minimization approach is introduced.

B. Projected Alternating Minimization

The projected alternating minimization divides the optimization in eq.(8) into three update steps: the gradient descent step and projection onto nonnegativity and normalization constraints. The gradient descent update for the sharp image as per (7) can be written as

$$\mathbf{S}^{t+1}(\mathbf{x}) = \mathbf{S}^t(\mathbf{x}) - \eta_s\Big(\mathbf{K}^t_-(\mathbf{x}) * \big(\mathbf{K}^t(\mathbf{x}) * \mathbf{S}^t(\mathbf{x}) - \mathbf{B}(\mathbf{x})\big) + \lambda\nabla\cdot\frac{\nabla\mathbf{S}^t(\mathbf{x})}{\|\nabla\mathbf{S}^t(\mathbf{x})\|_2}\Big) \tag{9}$$

for some step-size $\eta_s > 0$, $\mathbf{K}^t_-(\mathbf{x}) = \mathbf{K}^t(-\mathbf{x})$, and $\nabla\cdot$ denotes divergence. The first convolution in eq.(9) is performed linearly to be compatible with the dimensions. Similarly, the gradient descent update for the kernel can be written as

$$\mathbf{K}^{t+1/3}(\mathbf{x}) = \mathbf{K}^t(\mathbf{x}) - \eta_k\Big(\mathbf{S}^{t+1}_-(\mathbf{x}) * \big(\mathbf{K}^t(\mathbf{x}) * \mathbf{S}^{t+1}(\mathbf{x}) - \mathbf{B}(\mathbf{x})\big)\Big) \tag{10}$$

for some step-size $\eta_k > 0$ and $\mathbf{S}^{t+1}_-(\mathbf{x}) = \mathbf{S}^{t+1}(-\mathbf{x})$. The nonnegativity and normalization update steps can be written as follows

$$\mathbf{K}^{t+2/3}(\mathbf{x}) = \max\{\mathbf{K}^{t+1/3}(\mathbf{x}), 0\}, \tag{11}$$

$$\mathbf{K}^{t+1}(\mathbf{x}) = \frac{\mathbf{K}^{t+2/3}(\mathbf{x})}{\|\mathbf{K}^{t+2/3}(\mathbf{x})\|_1}. \tag{12}$$

The update steps are summarized in the Algorithm 1.

III. EXPERIMENTS

We consider 3D images of peppers and pears of size $384 \times 512 \times 3$ and $486 \times 732 \times 3$ respectively. A moving average filter of size 1×17 padded with zeros to make it 17×17 is applied on the sharp images resulting in blurred images of size $368 \times 496 \times 3$ and $470 \times 716 \times 3$. The blurred image and kernel size with parameters $\lambda_{\text{init}} = 0.5$, $\lambda_{\text{min}} = 0.0001$ are given as input to the Algorithm 1. The Algorithm 1 outputs the recovered sharp image and kernel. The true, blurred, and recovered images are shown in Fig. 1 and 3. To quantify the results, we use NMSE, PSNR, and SSIM as the metrics.

$$\text{NMSE}(\mathbf{S}, \widehat{\mathbf{S}}) = 10\log_{10}\big(\frac{\|\mathbf{S} - \widehat{\mathbf{S}}\|^2_{\text{F}}}{\|\widehat{\mathbf{S}}\|^2_{\text{F}}}\big) \tag{13}$$

$$\text{PSNR}(\mathbf{S}, \widehat{\mathbf{S}}) = 10\log_{10}\big(\frac{1}{\|\mathbf{S} - \widehat{\mathbf{S}}\|^2_{\text{F}}/(hs * ws)}\big) \tag{14}$$

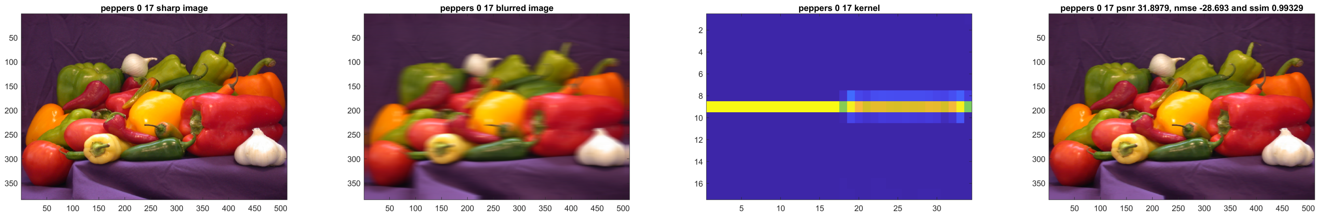
TABLE I: Performance evaluation for peppers and pears.

Image	NMSE	PSNR	SSIM
Peppers	-28.693	31.898	0.9933
Pears	-27.479	27.956	0.9821

Table I shows the recovered performance for peppers and pears images. The PAM algorithm takes roughly 3 hours to generate results for pears image of size $486 \times 732 \times 3$. Therefore, a pyramid scheme is used for quicker results.

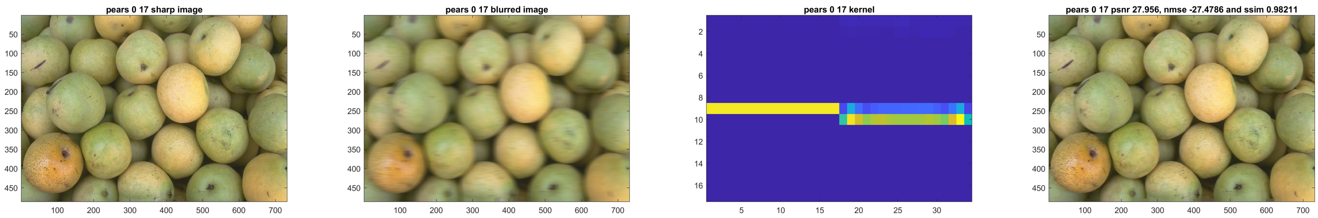
IV. REFERENCE

Perrone, Daniele, and Paolo Favaro. "A clearer picture of total variation blind deconvolution." IEEE transactions on pattern analysis and machine intelligence 38.6 (2015): 1041-1055.



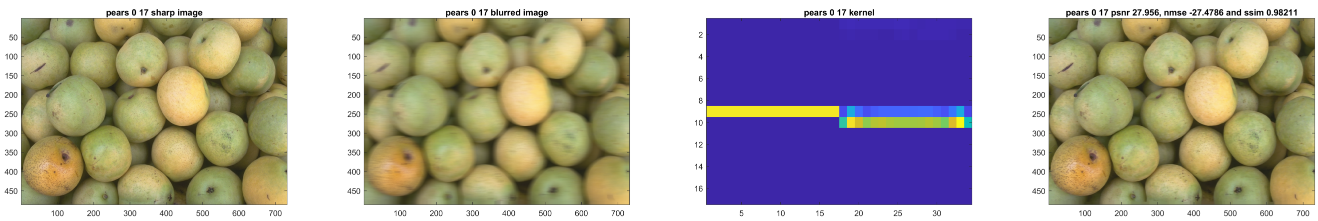
(a) (b) (c) (d)

Fig. 1: (from left to right) peppers sharp image, blurred image, true and estimated kernels, and recovered image.



(a) (b) (c) (d)

Fig. 2: (from left to right) pears sharp image, blurred image, true and estimated kernels, and recovered image.



(a) (b) (c) (d)

Fig. 3: (from left to right) pears sharp image, blurred image, true and estimated kernels, and recovered image.