Blind Image Deconvolution via Bayesian Inference

Wei-Jhe, Fong

National Tsing Hua University
Mathematics

December 1st, 2021

Outline

- Introduction
- Prior Estimation
  - Estimation with EM-Algorithm
  - Why EM-Algorithm Works?
- Blind Deconvolution
  - Maximum-a-Posterior and $\text{MAP}_{x,k}$
  - Difference between MAP and MAP_k
  - MAP_k with EM-Algorithm
- Non Blind Deconvolution
- Numerical Result
- Conclusion and Future Work

Introduction: Model

In our experience in taking images, there are great chance that one would accidentally shake his or her hands a little bit causing the image become fuzzy. Another common situation is that we often take the images without correct focus resulting unpleasant image quality.

Figure: Motion Blur with noise

Introduction: Model

In our experience in taking images, there are great chance that one would accidentally shake his or her hands a little bit causing the image become fuzzy. Another common situation is that we often take the images without correct focus resulting unpleasant image quality.

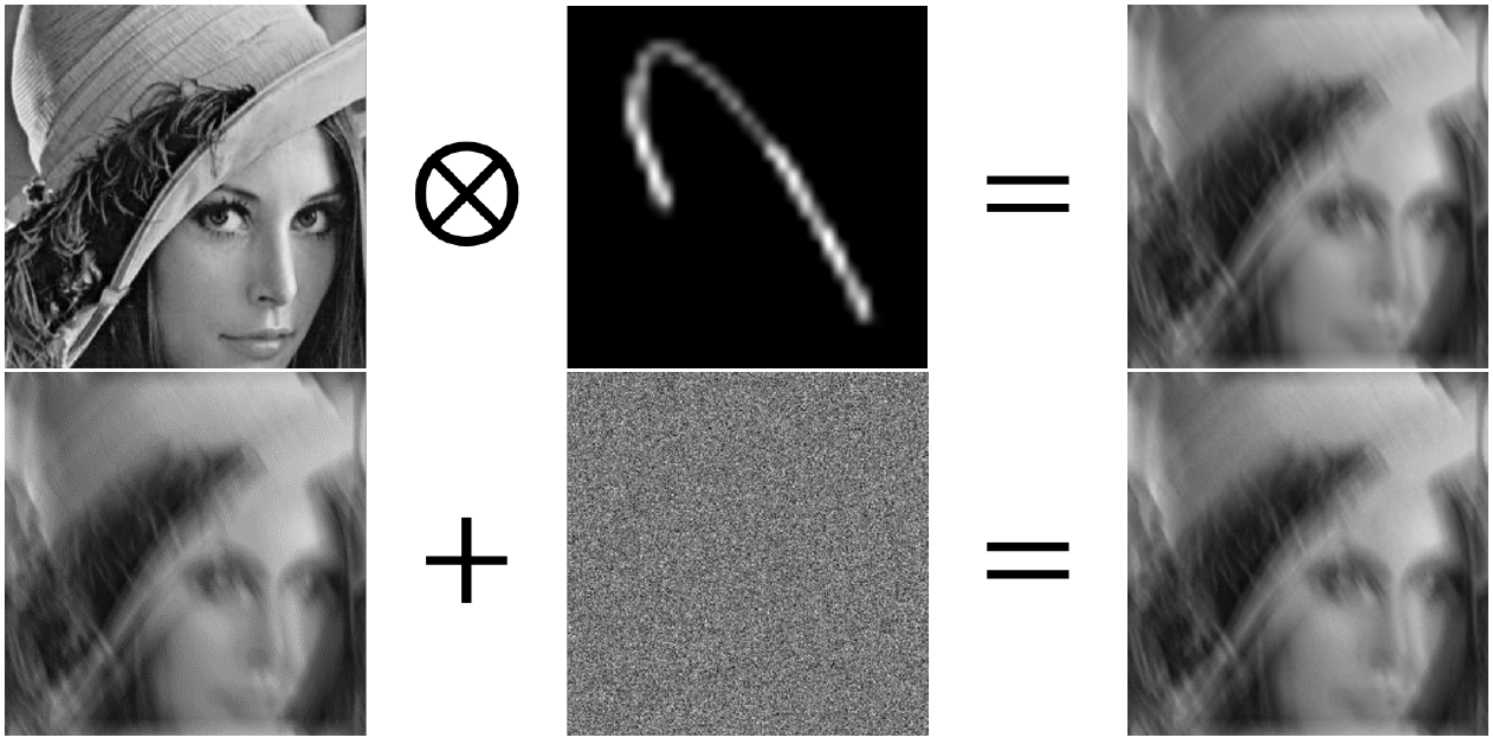
Figure: Defocus Blur with noise

Introduction: Model

With the blurred images, we have an image model:

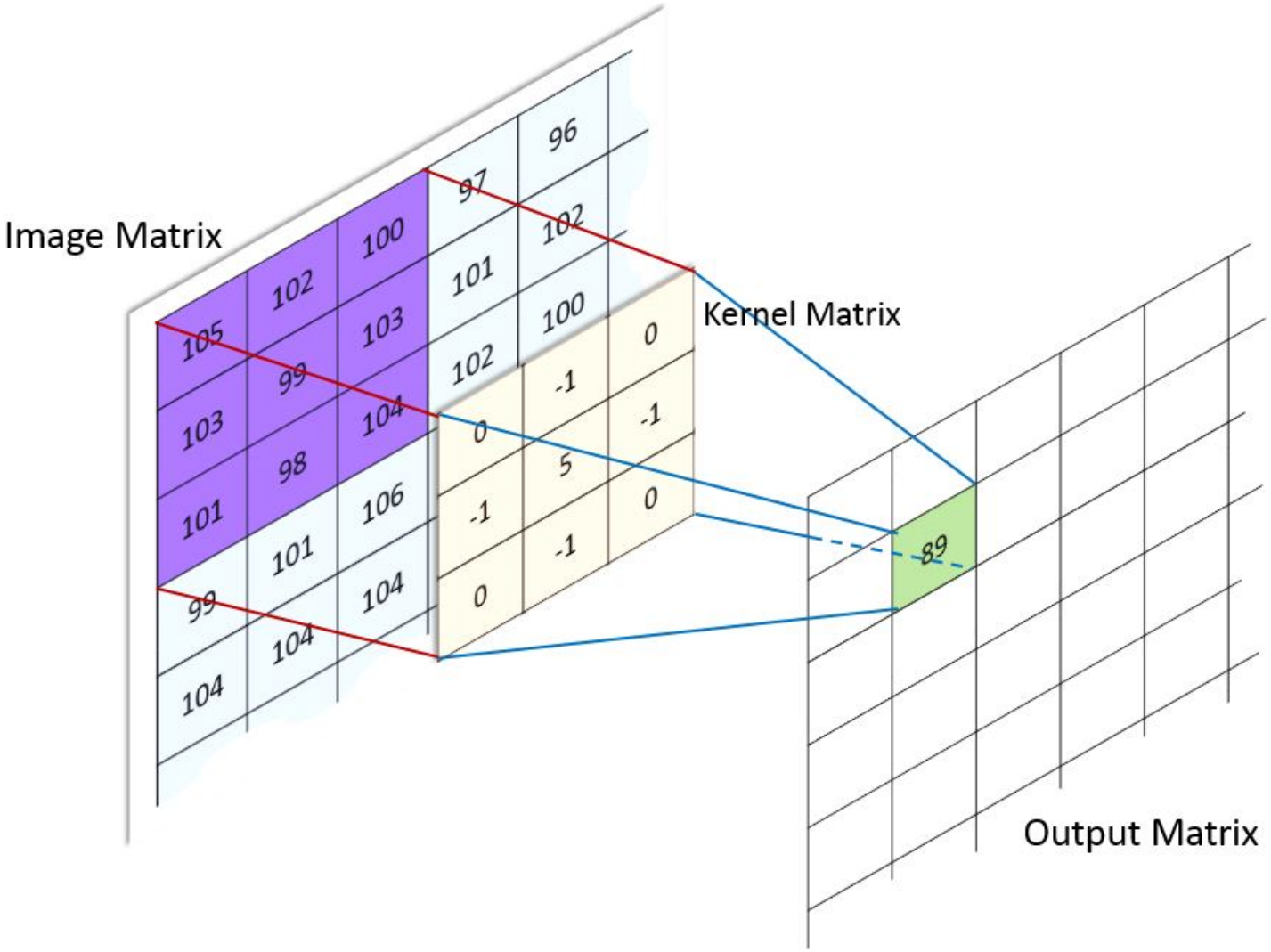
$$y = x \otimes k + n \tag{1}$$

where y is the blurred image, x is the sharp image, k is the blur kernel, n is the additive noise, and $\otimes$ is the convolution operator.



Introduction: Model

What is convolution?



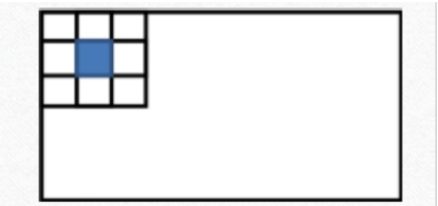
Introduction: Model

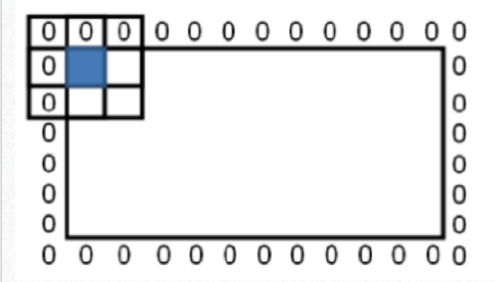
The process of convolution in mathematics can be described as

$$
\begin{aligned}
y(i,j) &= (x \otimes k)(i,j) + n(i,j) \\
&= \sum_{l_1=-\frac{m-1}{2}}^{\frac{m-1}{2}} \sum_{l_2=-\frac{t-1}{2}}^{\frac{t-1}{2}} x(i+l_1, j+l_2) k(\frac{m+1}{2} + l_1, \frac{t+1}{2} + l_2) + n(i,j)
\end{aligned}
\tag{2}
$$

where i, j are the index in the image y and $m+1, t+1$ are the length of kernel k

We can notice the borders of the image would have some problems.





Introduction: Probability Asumptions

The first assumption[1,2] is the noise of an image is gaussian, that is the prbability $p(y \mid x, k)$ can be describe as:

$$p(y \mid x, k) = \frac{1}{(\sqrt{2\pi}\eta)^N} \exp^{-\frac{1}{2\eta^2}\|x \otimes k - y\|^2} \tag{3}$$

Where N is the number of pixels and η is the deviation of the gaussain distribution.

[1] Rob Fergus et al. "Removing camera shake from a single photograph". In: *ACM SIGGRAPH 2006 Papers*. 2006, pp. 787–794.

[2] Anat Levin et al. "Understanding and evaluating blind deconvolution algorithms". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2009, pp. 1964–1971.

Introduction: Probability Assumptions

The second assumption is that the gradient histogram of natural image follows a mixture of J gaussian distribution[3]

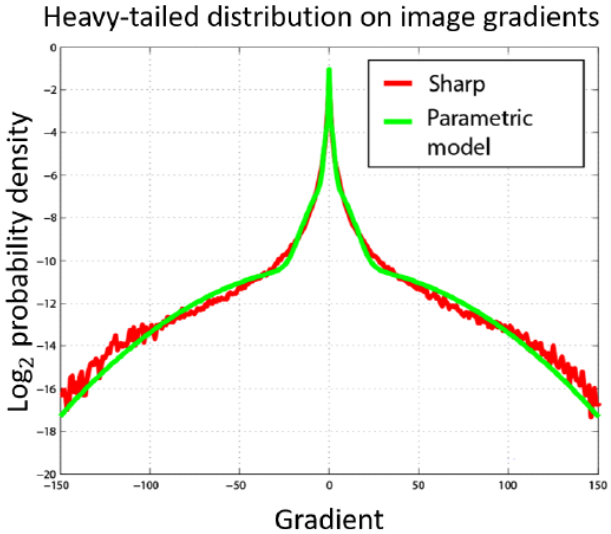
$$
\begin{aligned}
p(x) &= \prod_{i=1}^{N} p(x(i)) \\
p(x(i)) &= \prod_{l=1}^{2} \left(\sum_{j=1}^{J} \frac{\pi_j}{\sqrt{2\pi}\sigma_j} \exp^{-\frac{1}{2\sigma_j^2}|(g_l \otimes x)(i)|^2} \right)
\end{aligned}
\tag{4}
$$

where g_1, g_2 representing the horizontal and vertical derivative respectively, here we simply use $\{g_1, g_2\} = \{[-1, 1], [-1, 1]^T\}$.

[3] Anat Levin et al. "Efficient marginal likelihood optimization in blind deconvolution". In: *CVPR 2011*. IEEE. 2011, pp. 2657–2664.

Introduction: Probability Assumptions

The reason why we assume a mixture of gaussian prior for latent image is that there are some research[4][5] which have shown a specific behavior in gradients of natural images. That is, they follow the heavy-tailed distributions in their gradients.

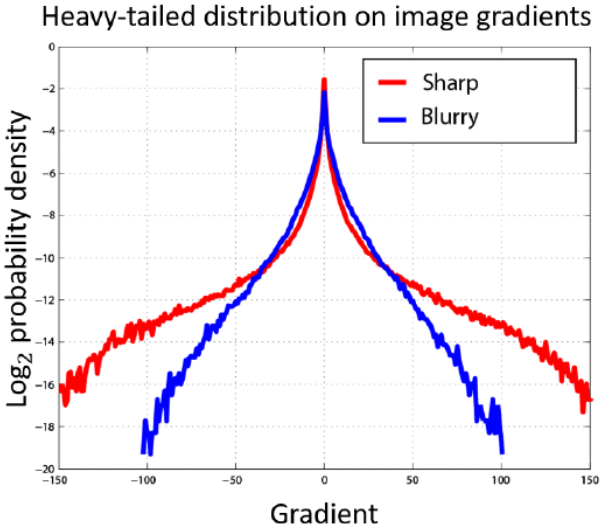


[4] Fergus et al., "Removing camera shake from a single photograph".

[5] David J Field. "What is the goal of sensory coding?" In: *Neural computation* 6.4 (1994), pp. 559–601.

Introduction: Probability Assumptions

The reason why we assume a mixture of gaussian prior for latent image is that there are some research[6][7] which have shown a specific behavior in gradients of natural images. That is, they follow the heavy-tailed distributions in their gradients.



[6] Fergus et al., "Removing camera shake from a single photograph".
[7] Field, "What is the goal of sensory coding?"

Introduction: Probability Assumptions

Sometimes performing deconvolution in gradient domain[8] is perferred. Hence we may also assume:

$$p(\nabla y|\nabla x, k) = \frac{1}{(\sqrt{2\pi}\eta)^N} \exp^{-\frac{1}{2\eta^2}\|\nabla x \otimes k - \nabla y\|^2} \tag{5}$$

the corresponding image model to (1) and prior (4) are

$$\nabla y = \nabla x \otimes k + n \tag{6}$$

$$p(\nabla x) = \prod_{i=1}^{N}\prod_{l=1}^{2}\left(\sum_{j=1}^{J}\frac{\pi_j}{\sqrt{2\pi}\sigma_j}\exp^{-\frac{1}{2\sigma_j^2}|(g_l \otimes x)(i)|^2}\right) \tag{7}$$

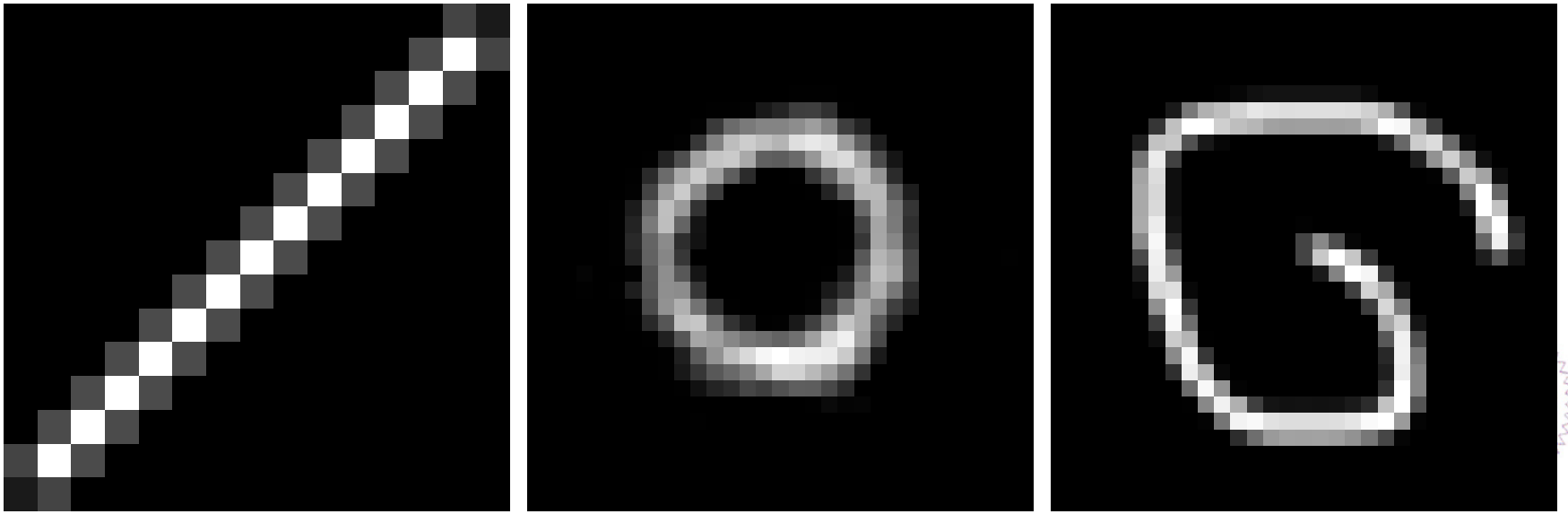
[8] Li Xu, Shicheng Zheng, and Jiaya Jia. "Unnatural l0 sparse representation for natural image deblurring". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2013, pp. 1107–1114.

Introduction: Probability Assumptions

Last, we typically assume a sparse prior $p(k)$ on the kernel k. The sparsity may be charaterize be L0-norm

$$p(k) = \exp^{-\beta \sum_i |k(i)|^0} \tag{8}$$

where $\beta \in \mathbb{R}$. Note that we also assume the kernel is non negative and always sum to 1 that is $\sum_{i=1}^{M} k(i) = 1$, where M is the number of pixels in the kernel.



Prior Estimation: Estimation with EM-Algorithm

We present the algorithm that fits the mixture of gaussian distribution in (4) from a number of flower images. The main algorithm is called the Expectation-Maximization Algorithm[9].

[9] Arthur P Dempster, Nan M Laird, and Donald B Rubin. "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1 (1977), pp. 1–22.

Prior Estimation: Estimation with EM-Algorithm

- Complete data $\mathcal{Z} = (\mathcal{X}, \mathcal{H})$.
- The observed data $\mathcal{X} = \{x_1, \cdots, x_n\}$ are the pixel-wise horizontal and vertical gradient values of a number of natural images. The probability $p(x, h; \Theta) := p(x|h; \Theta)p(h; \Theta) = \frac{1}{\sqrt{2\pi}\sigma_h} \exp^{-\frac{1}{2\sigma_h^2}|x|^2} \pi_h$
- The hidden data $\mathcal{H} = \{h_1, \cdots, h_n\}$ is a vector consist of n random variable s.t $h_i \in \{1, 2, \cdots, J\}$ and $p(h_i = j) = \pi_j$ (i.e $p(h_i) = \pi_{h_i}$)
- Let $\Theta_j = (\pi_j, \sigma_j)$, $j = 1, 2, \cdots, J$ be the unknown parameters. The goal of EM is:

$$\hat{\Theta} = \arg\max_{\Theta} \log P(\mathcal{X}; \Theta) \tag{9}$$

- Note that direct computation of MLE is impossible since the distribution is of mixture of gaussian[10].

[10] Christopher M Bishop. "Pattern recognition". In: *Machine learning* 128.9 (2006).

- **E-step:** We define

$$\mathcal{L}(\mathcal{X}, \mathcal{H}; \Theta) := \log\left(\prod_{i=1}^{n} p(x_i, h_i; \Theta)\right)$$

and

$$p(h_i; \mathcal{X}, \Theta_{old}) = \frac{p(h_i, x_i; \Theta_{old})}{p(x_i; \Theta_{old})} = \frac{p(h_i, x_i; \Theta_{old})}{\sum_{j=1}^{J} p(x_i, h = j; \Theta_{old})}$$

In E-step we compute:

$$Q(\Theta; \Theta_{old}) := E[\mathcal{L}(\mathcal{X}, \mathcal{H}; \Theta) \mid \mathcal{X}, \Theta_{old}] = \int \mathcal{L}(\mathcal{X}, H; \Theta) p(H; \mathcal{X}, \Theta_{old}) dH$$

$$= \sum_{i=1}^{n} \sum_{l=1}^{J} \log(\pi_l) p_{h_i}(l; \mathcal{X}, \Theta_{old})$$

$$+ \sum_{i=1}^{n} \sum_{l=1}^{J} \log\left(\frac{1}{\sqrt{2\pi}\sigma_l} \exp^{-\frac{1}{2\sigma_l^2}|x_i|^2}\right) p_{h_i}(l; \mathcal{X}, \Theta_{old})$$

- **M-step:** Find $\Theta_{new} = \arg\max_{\Theta} Q(\Theta; \Theta_{old})$
  - **Solve for π_l:** To find the expression for π_l, we introduce the Lagrange multiplier λ with the constraint that $\sum_l \pi_l = 1$, and solve the following equations:

$$\frac{\partial}{\partial \pi_s}\left[\sum_{i=1}^{n}\sum_{l=1}^{J}\log(\pi_l)p_{h_i}(l;\mathcal{X},\Theta_{old}) + \lambda\left(\sum_{l=1}^{J}\pi_l - 1\right)\right] = 0$$

So the solution is

$$\pi_s = \frac{1}{n}\sum_{i=1}^{n} p_{h_i}(s;\mathcal{X},\Theta_{old})$$

  - **Solve for σ_l:** Consider

$$\frac{\partial}{\partial \sigma_s^{-2}}\sum_{i=1}^{n}\sum_{l=1}^{J}\left(-\log\sqrt{2\pi} + \frac{1}{2}\log(\sigma_l^{-2} - \frac{|x_i|^2}{2\sigma_l^2})\right)p_{h_i}(l;\mathcal{X},\Theta_{old}) = 0.$$

So the solution is

$$\sigma_s^2 = \frac{\sum_{i=1}^{n}|x_i|^2 p_{h_i}(s;\mathcal{X},\Theta_{old})}{\sum_{i=1}^{n} p_{h_i}(s;\mathcal{X},\Theta_{old})}$$

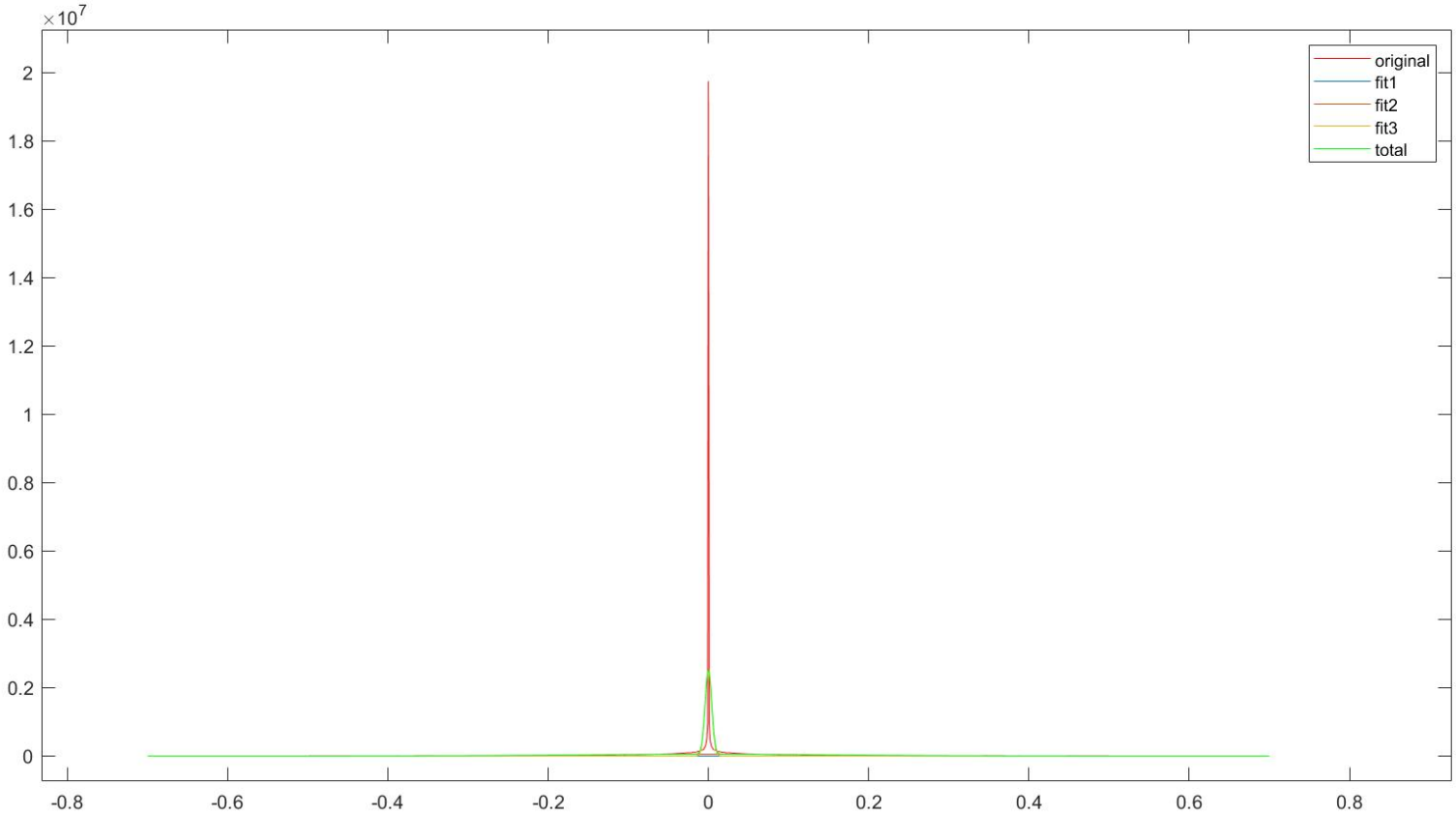
Prior Estimation: Why it works?

Observe that

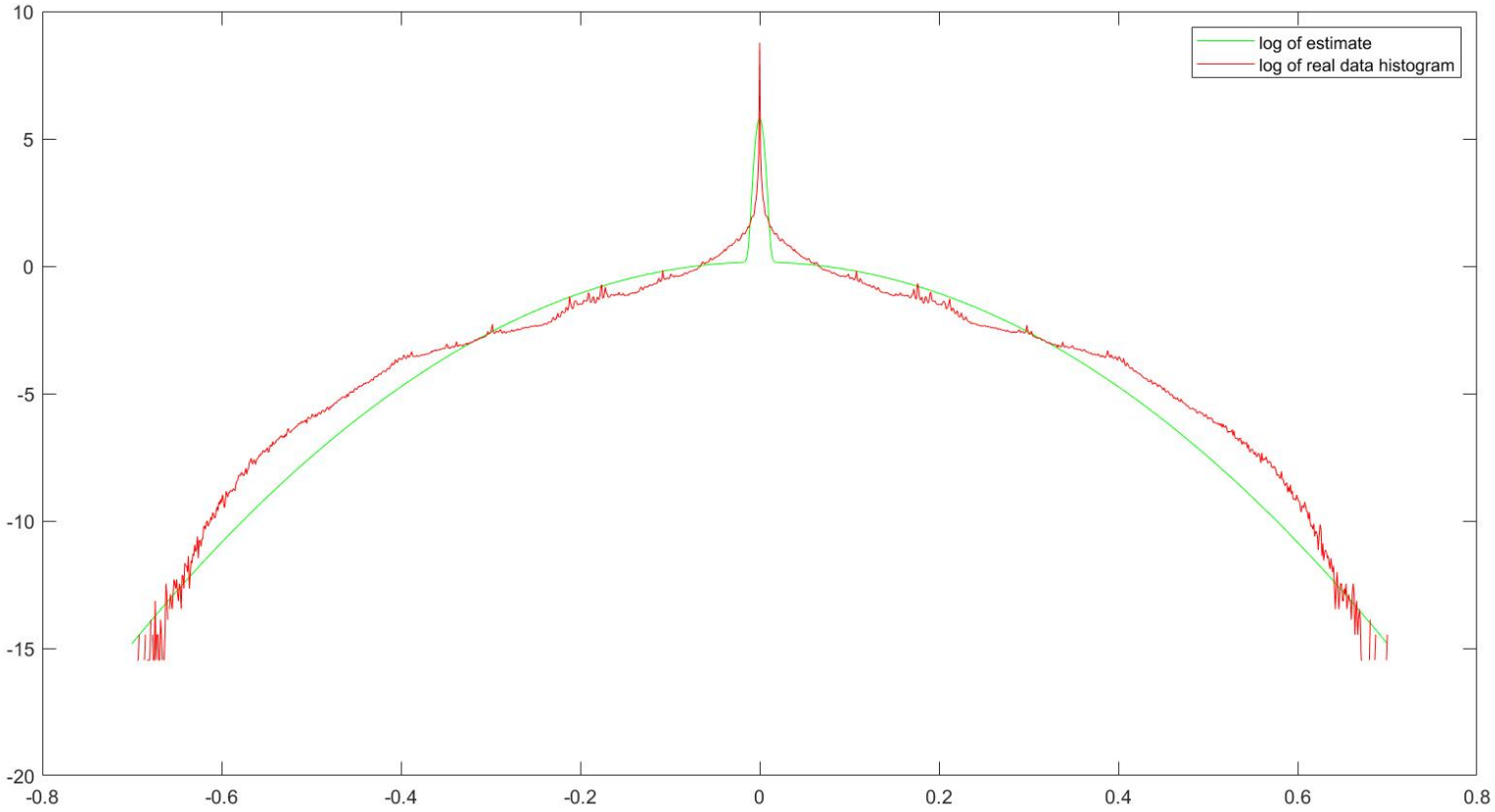
$$
\begin{aligned}
\mathcal{L}(\mathcal{X};\Theta) &= \log P(\mathcal{X};\Theta) = \log \int P(\mathcal{X},H;\Theta)dH \\
&= \log \int \frac{P(\mathcal{X},H;\Theta)}{P(H|\mathcal{X};\Theta_{old})} P(H|\mathcal{X};\Theta_{old})dH \\
&= \log E\left[\frac{P(\mathcal{X},\mathcal{H};\Theta)}{P(\mathcal{H}|\mathcal{X};\Theta_{old})} \mid \mathcal{X},\Theta_{old}\right] \\
&\geq E\left[\log\left(\frac{P(\mathcal{X},\mathcal{H};\Theta)}{P(\mathcal{H}|\mathcal{X};\Theta_{old})}\right) \mid \mathcal{X},\Theta_{old}\right] \\
&= Q(\Theta;\Theta_{old}) - E[\log(P(\mathcal{H}|\mathcal{X};\Theta_{old})) \mid \mathcal{X},\Theta_{old}]
\end{aligned}
$$

Therefore any value of Θ that increases $Q(\Theta;\Theta_{old})$ must also increase $\mathcal{L}(\mathcal{X};\Theta)$ beyond $\mathcal{L}(\mathcal{X};\Theta_{old})$.

Prior Estimation: Result



Prior Estimation: Result



Blind Deconvolution: Maximum-a-Posterior and $\text{MAP}_{x,k}$

- The most intuitive solution: finding the MLE of $(\nabla x, k)$

$$(\nabla \hat{x}, \hat{k}) = \arg\max_{\nabla x, k} p(\nabla y | \nabla x, k)$$

  it doesn't take account the prior informations for image and kernel.
- Therefore we consider Maximum-a-Posterior Estimation of $(\nabla x, k)$

$$(\hat{\nabla x}, \hat{k}) = \arg\max_{\nabla x, k} p(\nabla x, k | \nabla y) = \arg\max_{\nabla x, k} p(\nabla y | \nabla x, k) p(\nabla x) p(k) \tag{10}$$

  In Fergus's research[11], they find that optimizing the cost function of MAP with conjugate gradient search tend to get trivial solution.
- The objective function (10) tends to minimize all gradients into zeros while we expect some large gradients in natural image. On the other hand if we increasing the weight on data fitting term, the result tends to be the no blur explaination since it has zero value in the data fitting term.

[11] Fergus et al., "Removing camera shake from a single photograph".

Blind Deconvolution: Maximum-a-Posterior and $\text{MAP}_{x,k}$

To avoid the problems cause by direct optimization of MAP cost function, the author[12] adopt a variational Bayesian approach.

1. Approximate $p(\nabla x, k|\nabla y)$ with $q(\nabla x, k)$ where $q(\nabla x, k)$ is chosen from some simply paramatric family
2. Compute the kernel k with maximum marginal probability. That is

$$\hat{k} = \arg\max_{k} \int q(\nabla x, k)\ d(\nabla x) \tag{11}$$

In pratice, they assume $q(\nabla x, k) = q(\nabla x)q(k)$ where $q(\nabla x)$ is a gaussain distribution , and $q(k)$ is a rectified Gaussian distribution.

[12] Fergus et al., "Removing camera shake from a single photograph".

Blind Deconvolution: Difference between MAP and MAP_k

Levin[13] look into the success of Fergus's research.

- There is a strong asymmetry between the dimensionalities of x and k.
- Instead of finding (x, k) at the same time, consider

$$\hat{k} = \arg\max_k p(k|y) \tag{12}$$

follows by a non blind deconvolution.

[13] Levin et al., "Efficient marginal likelihood optimization in blind deconvolution".

Blind Deconvolution: Difference between MAP and MAP_k

- Suppose 1D blind deconvolution problem: $y = kx + n$.
- Assume $n \sim N(0, \eta^2)$, $x \sim N(0, \sigma^2)$ and k follows a uniform distribution.(i.e p(k) is a constant)
- $p(x, k|y) \propto \exp^{-\frac{1}{2\eta^2}(kx-y)^2 - \frac{x^2}{2\sigma^2}}$ which increases when $x \to 0$ and $k \to +\infty$.
- On the other hand,

$$\hat{k} = \arg\max_k \int p(x, k|y)\ dx = \arg\max_k \int \exp^{-\frac{1}{2\eta^2}(kx-y)^2 - \frac{x^2}{2\sigma^2}}\ dx$$

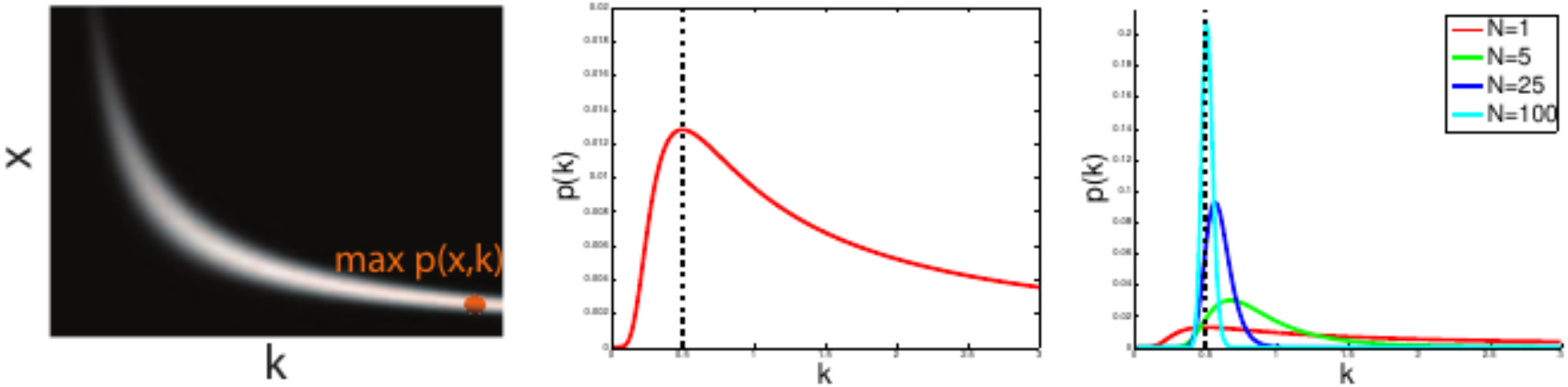
the cost function of MAP_k no longer increases when $k \to \infty$.

Blind Deconvolution: Difference between MAP and MAP_k

Claim

[a] *Suppose the image x is arbitrary large, sampling from the prior $p(x)$ and $y = k \otimes x + n$. Then the cost function of MAP_k (i.e $p(y|k)$) is maximized by the true kernel k^*. Moreover, if $\arg\max_k p(y|k)$ is unique, $p(k|y)$ approaches a delta function.*

[a] Levin et al., "Understanding and evaluating blind deconvolution algorithms".



(a) $p(x, k|y)$ (b) $p(k|y_1)$ (c) $p(k|y_1, ..y_N)$

Blind Deconvolution: Difference between MAP and MAP_k

Another way to understand the difference between MAP_k and MAP:

- Define $L(x, \hat{x}, k, \hat{k})$ to be the loss function. The Bayes estimator of unknown parameter $\hat{\Theta} = (\hat{x}, \hat{k})$ is:

$$(\hat{x}, \hat{k}) = \arg\min_{\bar{x}, \bar{k}} \int \int p(x, k|y) L(x, \bar{x}, k, \bar{k}) dx dk$$

- MAP estimator is equivalent to define the loss function as:

$$L(x, \hat{x}, k, \hat{k}) = 1 - \delta((\hat{x}, \hat{k}) - (x, k))$$

- Assume $p(k|y)$ has a unique maxima then MAP_k is equivalent to define the loss function as:

$$L(x, \hat{x}, k, \hat{k}) = |x - \hat{x}|^2 + |k - \hat{k}|^2$$

Blind Deconvolution: MAP_k with EM-Algorithm

Suppose there is only one component in the mixture of gaussian prior, then we can implement a standard EM-algorithm to find the solution:

1. **E-step:** Fix k, y, we calculate

$$q(x) = p(x|y,k) = p(x,y|k)/p(y) \propto p(x,y|k) \tag{13}$$

2. **M-Step:** Find $\hat{k}$ s.t

$$\hat{k} = \arg\min_k E_q(\|k \otimes x - y\|^2) - \log p(k) \tag{14}$$

$$= \left(\int q(x) \|k \otimes x - y\|^2 dx \right) - \log p(k) \tag{15}$$

Blind Deconvolution: MAP_k with EM-Algorithm

- **E-step:**

$$
\begin{aligned}
&-\log(p(x,y|k)) = -\log(p(y|x,k)) - \log(p(x)) \\
&= \frac{1}{2\eta^2}\|k \otimes x - y\|^2 + N\log(\sqrt{2\pi}\eta) \\
&+ \sum_{l=1}^{2}\left(\frac{1}{2\sigma_1^2}\|g_l \otimes x\|^2 - \log(\pi_1) + \log(\sqrt{2\pi}\sigma_1)\right) \\
&= \frac{1}{2\eta^2}\|k \otimes x - y\|^2 + \sum_{l=1}^{2}\frac{1}{2\sigma_1^2}\|g_l \otimes x\|^2 + c
\end{aligned} \tag{16}
$$

where $c = N\log(\sqrt{2\pi}\eta) + 2\log(\sqrt{2\pi}\sigma_1) - 2\log(\pi_1)$ is the constant.

Blind Deconvolution: MAP_k with EM-Algorithm

- **E-Step:** Equation (22) can be simplified further more:

$$-\log\ q(x) \propto \frac{1}{2}x^T A_k x - b_k^T x + c \tag{17}$$

where

$$\begin{cases} A_k = \dfrac{1}{\eta^2} T_k^T T_k + \dfrac{1}{\sigma^2} \displaystyle\sum_{l=1}^{2} T_{g_l}^T T_{g_l} \\ b_k = \dfrac{1}{\eta^2} T_k^T y \end{cases} \tag{18}$$

Hence we can see that $q(x)$ is also a Gaussian with mean μ and covariance C to be

$$\begin{cases} \mu = A_k^{-1} b_k \\ C = A_k^{-1} \end{cases} \tag{19}$$

Blind Deconvolution: MAP_k with EM-Algorithm

- **M-Step:**

$$E_q(\|k \otimes x - y\|^2) = \int q(x)\|k \otimes x - y\|^2 dx = \frac{1}{2}k^T \bar{A} k - \bar{b}^T k + c \quad (20)$$

For a fixed x, the convolution error is quadratic in k and therefore can be written as

$$\|k \otimes x - y\|^2 = k^T A_x k - b_x^T k$$

If k is an $m \times m$ kernel and $M = m^2, A_x$ is an $M \times M$ matrix representing the covariance of all $m \times m$ windows in x, and b_x the correlation with y :

$$A_x(i_1, i_2) = \sum_i x(i + i_1)\, x(i + i_2), \quad b_x(i_1) = \sum_i x(i + i_1)\, y(i)$$

where i sums over all image pixels, and i_1, i_2 are kernel indexes

Blind Deconvolution: MAP_k with EM-Algorithm

- **M-Step:**

$$E_q(\|k \otimes x - y\|^2) = \int q(x)\|k \otimes x - y\|^2 dx = \frac{1}{2\eta^2} k^T \bar{A} k - \bar{b}^T k + c \quad (21)$$

where

$$\begin{cases} \bar{A}(i_1, i_2) = \left(\sum_{i=1}^{N} \mu(i + i_1)\mu(i + i_2) \right) + C(i + i_1, i + i_2) \\ \bar{b} = \sum_{i=1}^{N} \mu(i + i_1) y(i) \end{cases} \quad (22)$$

Blind Deconvolution: MAP_k with EM-Algorithm

- **M-Step:** After calculating for $E_q(\|k \otimes x - y\|^2)$, we consider

$$F(k) = E_q(\|k \otimes x - y\|^2) - \log p(k) = \frac{1}{2\eta^2}\left(k^T \bar{A} k - \bar{b}^T k\right) + \beta \|k\|_0$$

The minimization could be solved by adding auxiliary value r to the objective function[14] and modify the cost function to

$$(\hat{k}, \hat{r}) = \arg\min_{k,r} \frac{1}{2\eta^2}\left(k^T \bar{A} k - \bar{b}^T k\right) + \frac{1}{2}\beta \|k - r\|^2 + \lambda \|r\|_0 \quad (23)$$

where r is the auxiliary value, and λ is the weight respect to r.

[14] Xu, Zheng, and Jia, "Unnatural l0 sparse representation for natural image deblurring".

Blind Deconvolution: MAP_k with EM-Algorithm

- **M-Step:** The minimization is solved by coordinate decsending.
  - **Solve for r:**

$$r = \begin{cases} k & \text{, if } |k|^2 \geq \dfrac{2\lambda}{\beta} \\ 0 & \text{, otherwise} \end{cases}$$

  - **Solve for k:**

$$\hat{k} = \underset{k>0, \|k\|_1=1}{\arg\min} \frac{1}{2\eta^2} \left(k^T(\bar{A} + 2\eta^2\beta 1_M)k\right) - (\bar{b}^T + \beta r^T)k$$

Non Blind Deconvolution: Wiener

The most intuitive way for Non Blind Deconvolution is finding the wiener filter w that minimize the mean square error:

$$\hat{x} = \arg\min_{\bar{x}} E\left[\|x - \bar{x}\|_2^2\right]$$
$$y = \hat{k} \otimes x + n, \ \bar{x} = w \otimes y$$

Where n follows a gaussian distribution. By parseval theorem we can take the minimization into frequency domain. Considering the fourier transform $Y, \hat{K}, X, N$ of $y, \hat{k}, x, n$, Under the independent assumption[15] of random variables X, N, the solution of W is

$$W(u,v) = \frac{K^*(u,v)}{|K(u,v)|^2 + \mathrm{NSR}(N,X)}$$

where the constant NSR interpreted as the Noise to Signal ratio between N and X,

[15] Rafael C Gonzalez and Richard E Woods. "Digital image processing 4th edition, global edition". In: (2018).

Non Blind Deconvolution:TV-L1

In order to achieve high robustness, avoid increasing the noise of the blur image and the ring effect, the researchers proposed a model with sparse prior on noise[16].
The objective function in the research is proposed as

$$\hat{x} = \arg\min_{\bar{x}} \|\hat{k} \otimes \bar{x} - y\|_1 + \lambda \|\nabla \bar{x}\|_1 \tag{24}$$

To solve the l_1 norm minimization, we adopted alternating minimization method basing on half-quadratic splitting.

$$(\hat{x}, \hat{w}, \hat{v}) = \arg\min_{\bar{x},w,v} \frac{1}{2\beta}\|\hat{k}\otimes\bar{x}-y-v\|_2^2+\frac{1}{2\theta}\|\nabla\bar{x}-w\|_2^2+\|v\|_1+\lambda\|w\|_1 \tag{25}$$

where $v = \mathbb{R}^{n_1 \times n_2}$, $w = (w_x, w_y)$ and $\theta, \beta \in \mathbb{R}$

[16] Li Xu and Jiaya Jia. "Two-phase kernel estimation for robust motion deblurring". In: *European conference on computer vision*. Springer. 2010, pp. 157–170.

Non Blind Deconvolution:TV-L1

- **Solve for $\hat{v}$:** The objective function w.r.t v seperated from (25) is

$$\hat{v} = \arg\min_{v} \frac{1}{2}\|v - (\hat{k} \otimes \hat{x} - y)\|_2^2 + \beta\|v\|_1 \tag{26}$$

We have the solution

$$\hat{v} = \text{sign}(\hat{k} \otimes \hat{x} - y)\max(\|\hat{k} \otimes \hat{x} - y\| - \beta, 0) \tag{27}$$

- **Solve for $\hat{w}$:** Similarly for w, we have the objective function

$$\hat{w} = \arg\min_{w} \frac{1}{2}\|w - \nabla\hat{x}\|_2^2 + \theta\lambda\|w\|_2 \tag{28}$$

The solution $\hat{w} = (\hat{w}_x, \hat{w}_y)$ can be written as

$$\begin{cases} \hat{w}_x = \dfrac{\partial_x \hat{x}}{\|\nabla\hat{x}\|_2}\max(\|\nabla\hat{x}\|_2 - \theta\lambda, 0) \\ \hat{w}_y = \dfrac{\partial_y \hat{x}}{\|\nabla\hat{x}\|_2}\max(\|\nabla\hat{x}\|_2 - \theta\lambda, 0) \end{cases} \tag{29}$$

where $\|\nabla\hat{x}\|_2 = \sqrt{(\partial_x\hat{x})^2 + (\partial_y\hat{x})^2}$

Non Blind Deconvolution:TV-L1

- **Solve for $\hat{x}$:** Last, for solving $\hat{x}$, we have the objective function

$$\hat{x} = \arg\min_{\bar{x}} \|\hat{k} \otimes \bar{x} - y - v\|_2^2 + \frac{\beta}{\theta}\|\nabla\bar{x} - w\|_2^2 \tag{30}$$

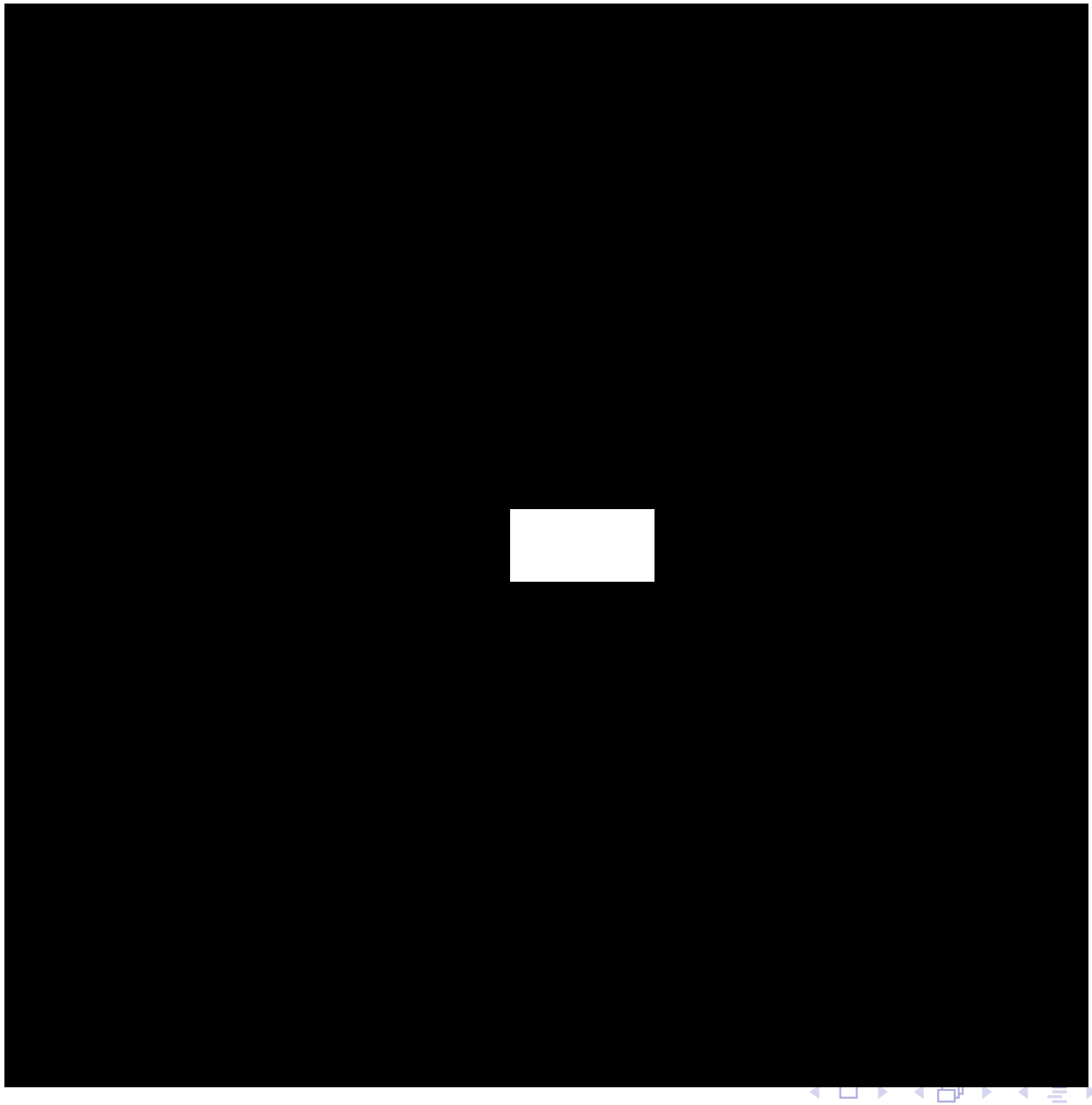
We can see that the cost function is of 2-norm, by Parseval theorem we can solve the minimization more easily in frequency domain. Therefore the objective function becomes

$$\hat{X} = \arg\min_{\bar{X}} \|\hat{K}\bar{X} - Y - V\|_2^2 + \frac{\beta}{\theta}\|\nabla\bar{X} - W\|_2^2 \tag{31}$$

Hence the solution is

$$\hat{X} = \frac{\hat{K}(Y+V) + \frac{\beta}{\theta}\left(\tilde{\nabla}^* W\right)}{\left(\hat{K}^*\hat{K} + \frac{\beta}{\theta}\tilde{\nabla}^*\tilde{\nabla}\right)} \tag{32}$$

Numerical Result



Numerical Result

Numerical Result

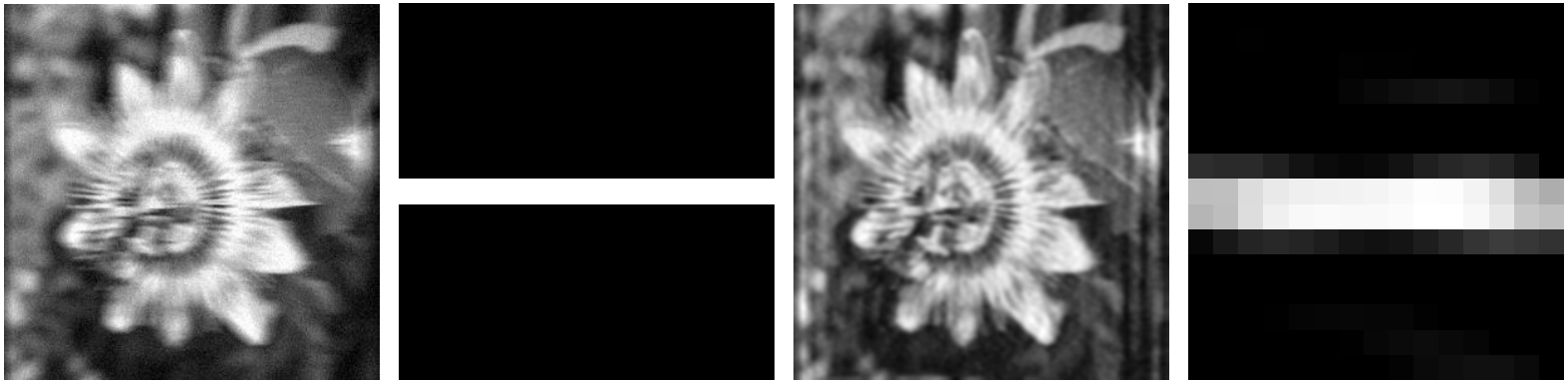


Figure: **(a)(b)**. Blur image(1) with the horizontal kernel and add gaussian noise. **(c)(d)**. Corresponding deblur image of (a) with the estimated kernel.

Numerical Result

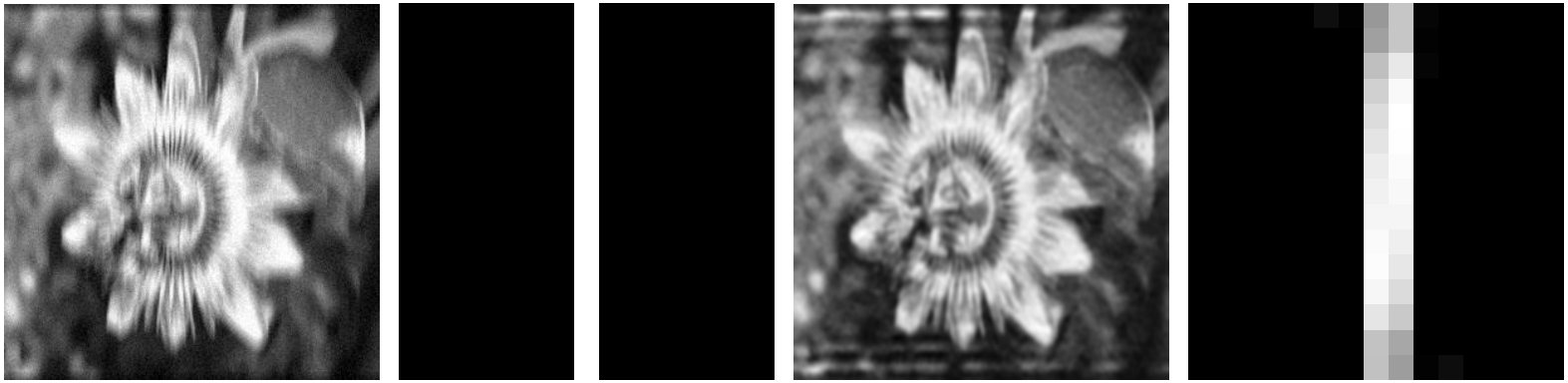


Figure: **(a)(b)**. Blur image(1) with the vertical kernel and add gaussian noise. **(c)(d)**. Corresponding deblur image of (a) with the estimated kernel.

Numerical Result

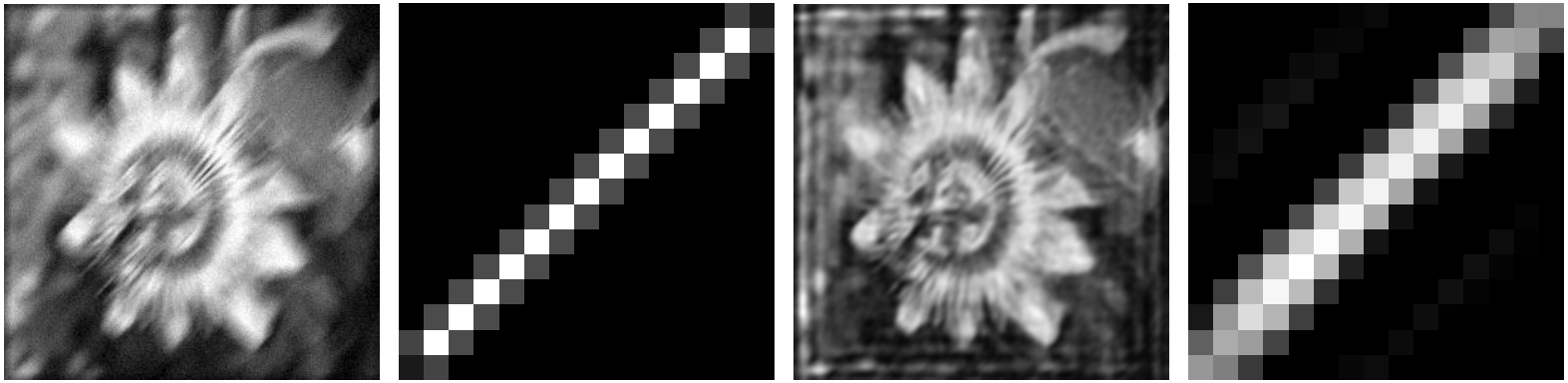


Figure: **(a)(b)**. Blur image(1) with the 45 degree line kernel and add gaussian noise. **(c)(d)**. Corresponding deblur image of (a) with the estimated kernel.

Numerical Result

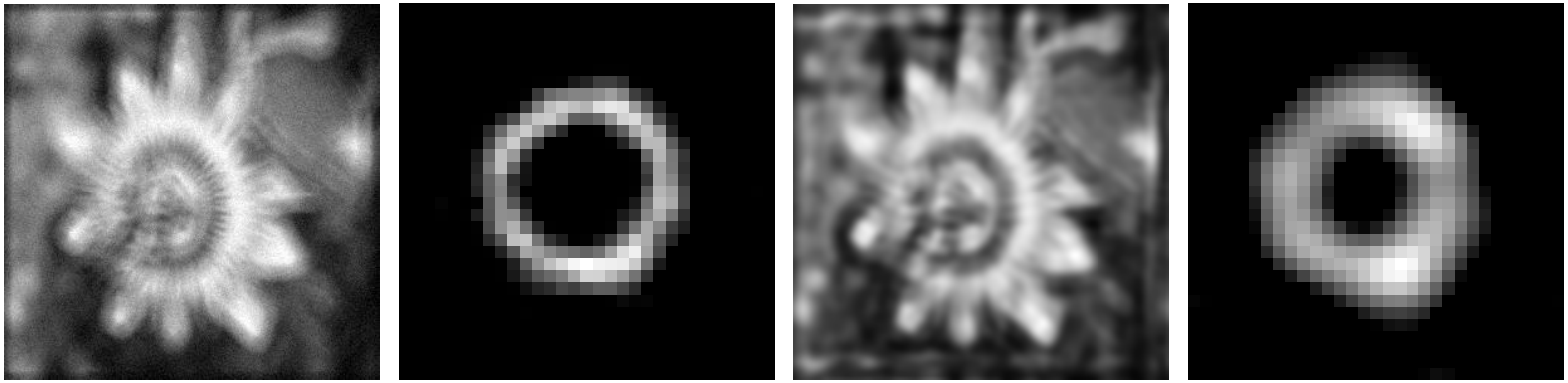


Figure: **(a)(b)**. Blur image(1) with the ring kernel and add gaussian noise. **(c)(d)**. Corresponding deblur image of (a) with the estimated kernel.

Numerical Result

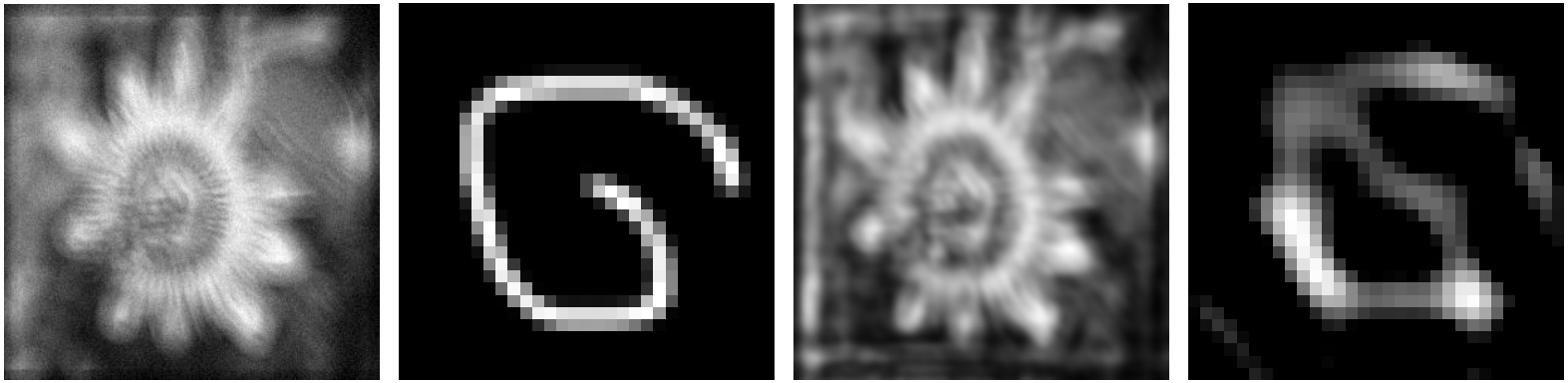
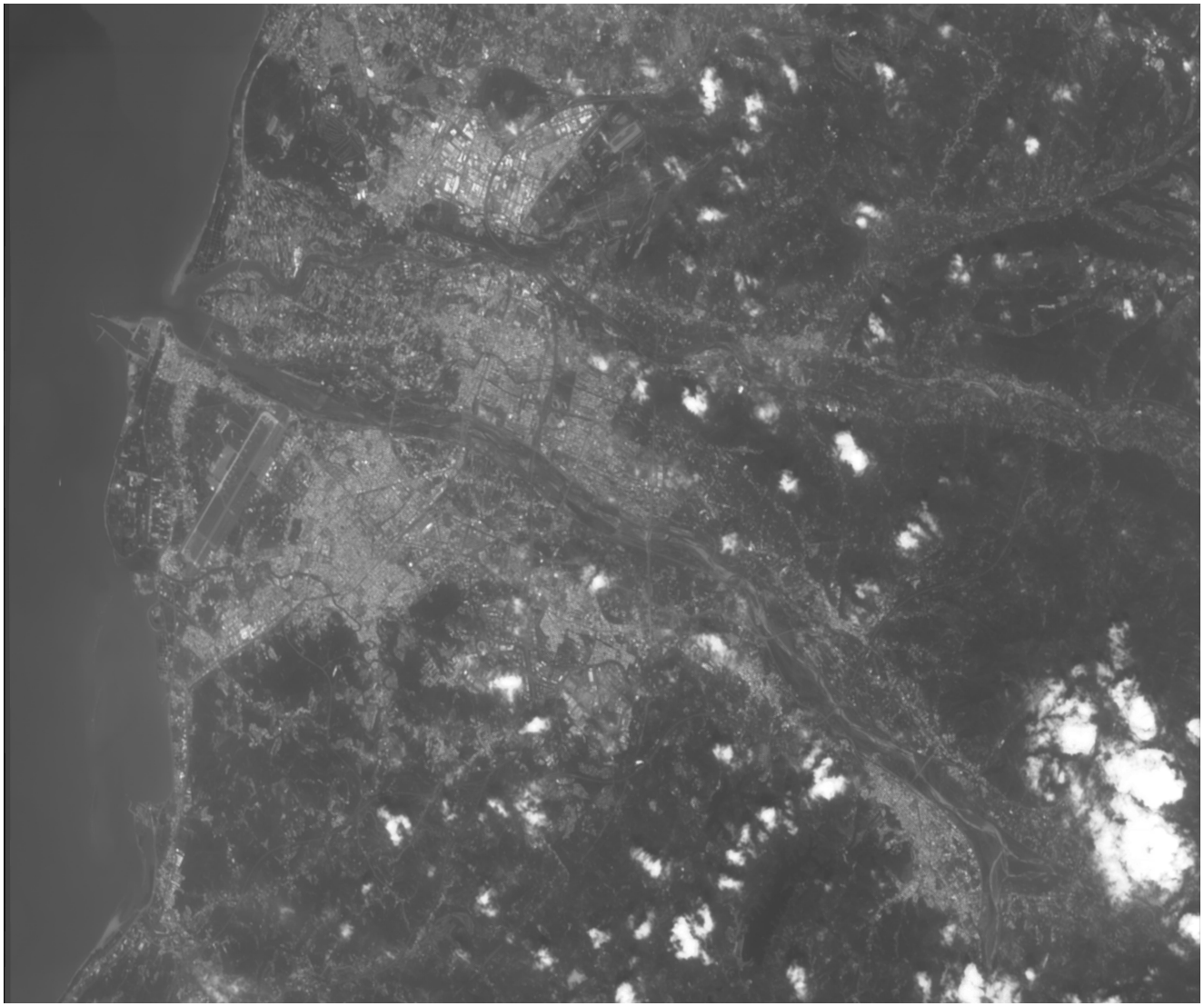


Figure: **(a)(b)**. Blur image(1) with a curve like kernel and add gaussian noise. **(c)(d)**. Corresponding deblur image of (a) with the estimated kernel.

Conclusion and Future Work

- What is a good prior?
- Different patches in an image may have different estimated kernel.
- Combine with machine learning?

Numerical Result



Numerical Result

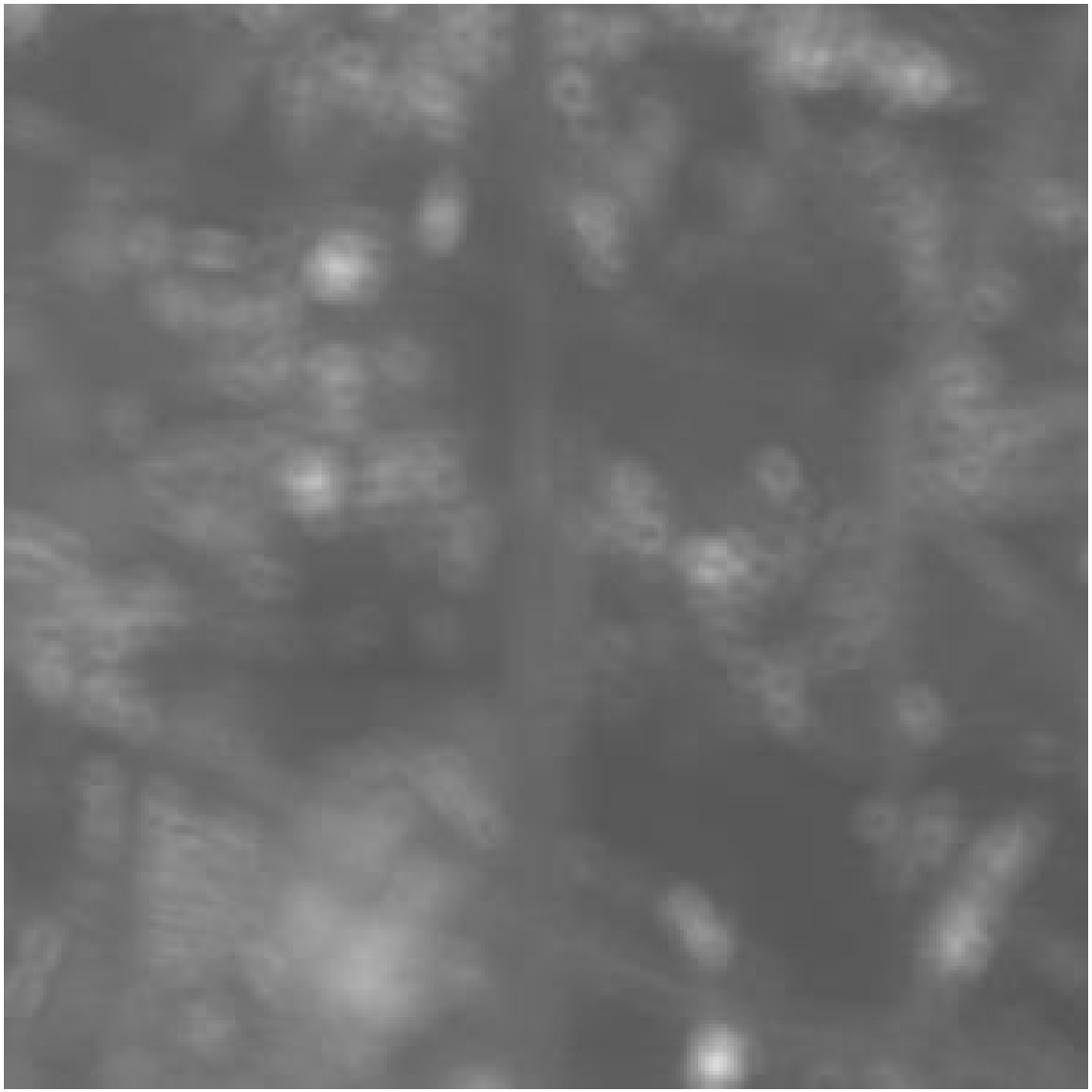


Figure: Patch 01

Numerical Result

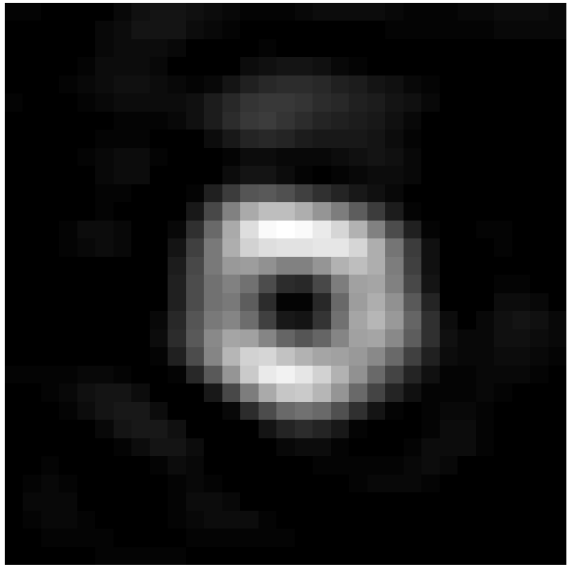


Figure: Kernel Estimation of patch 01

Numerical Result

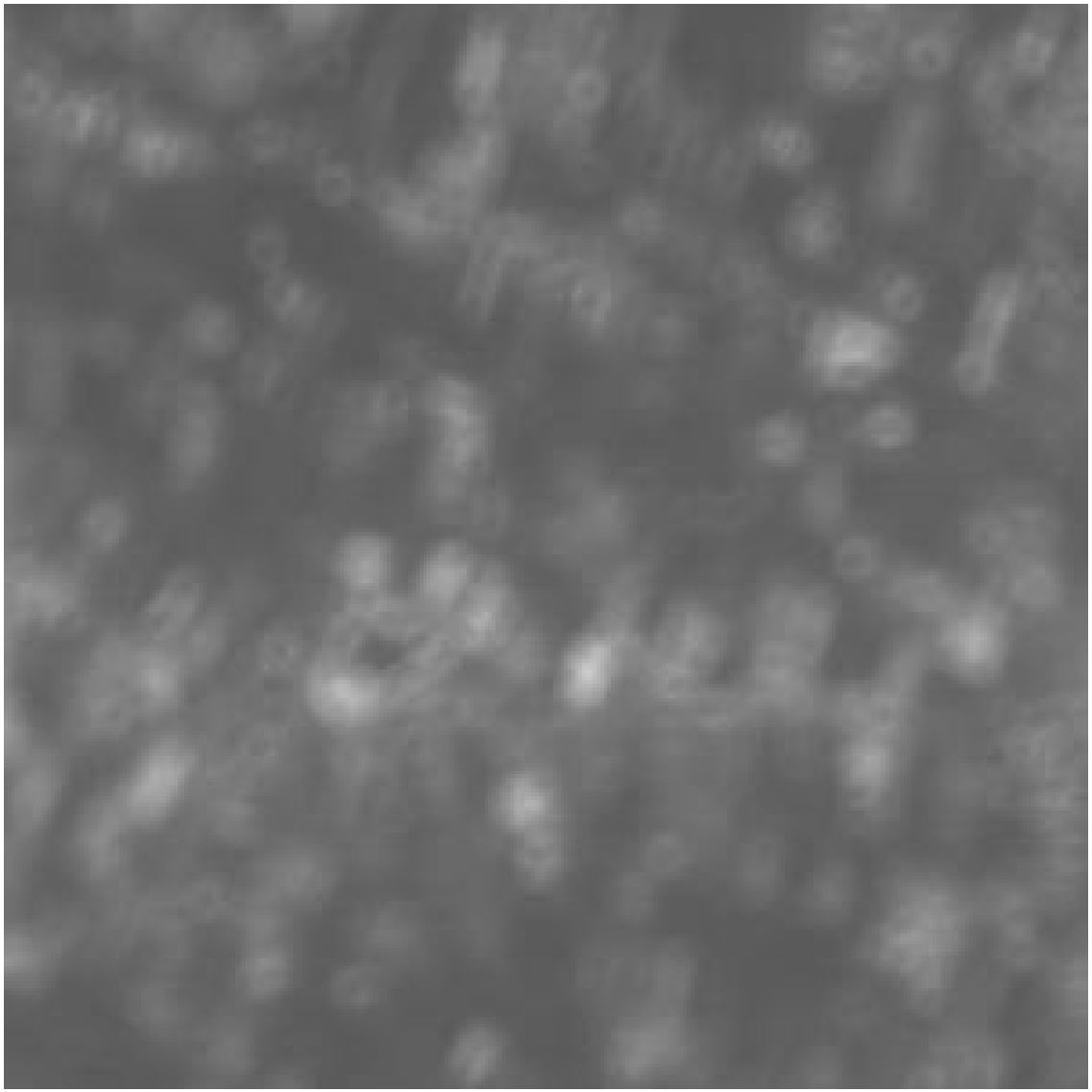


Figure: Patch 02

Numerical Result

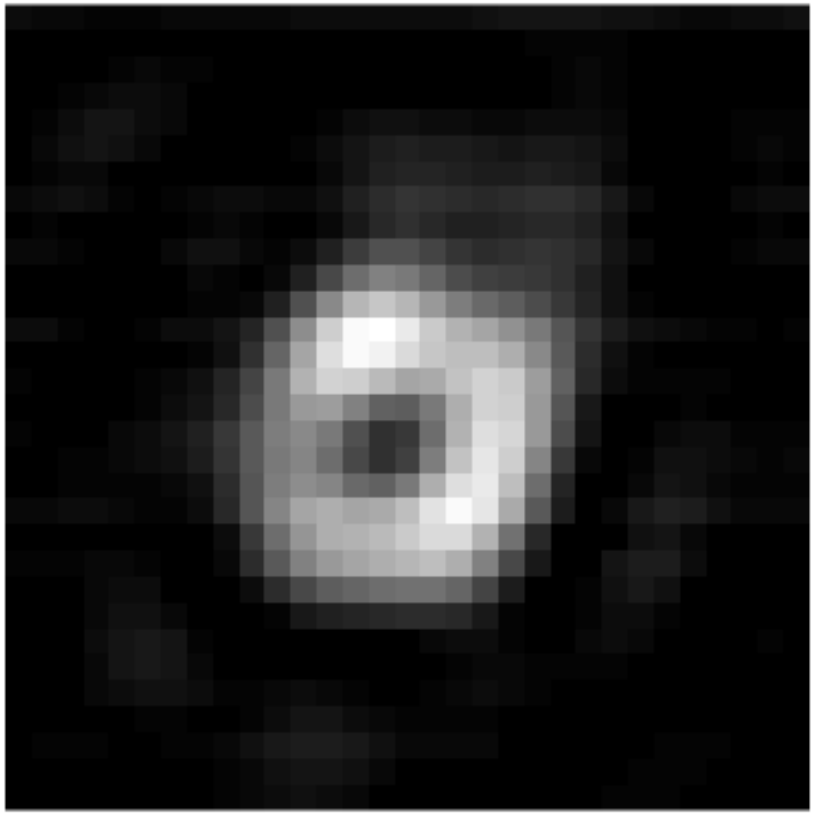


Figure: Kernel Estimation of patch 02

Numerical Result

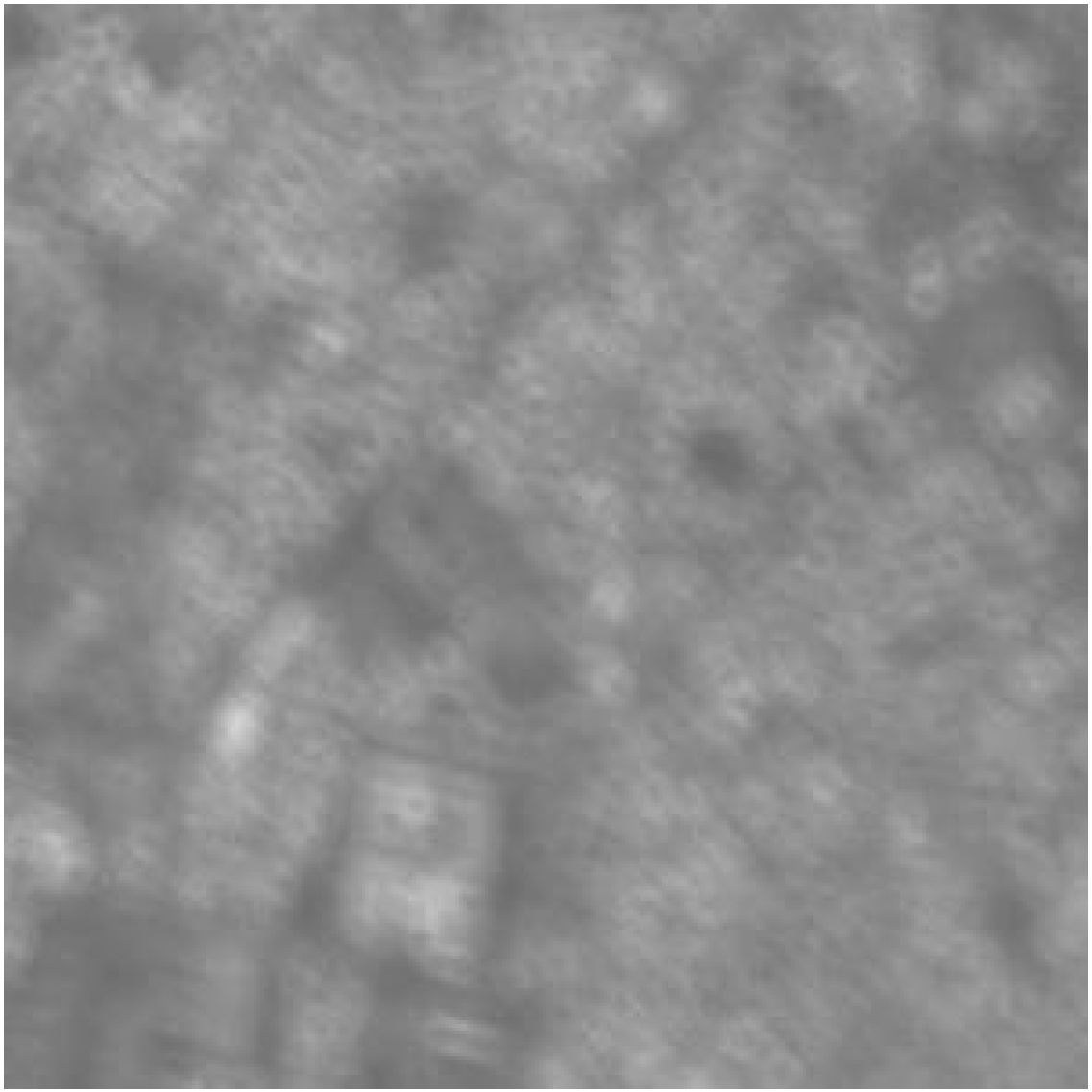


Figure: Patch 03

Numerical Result

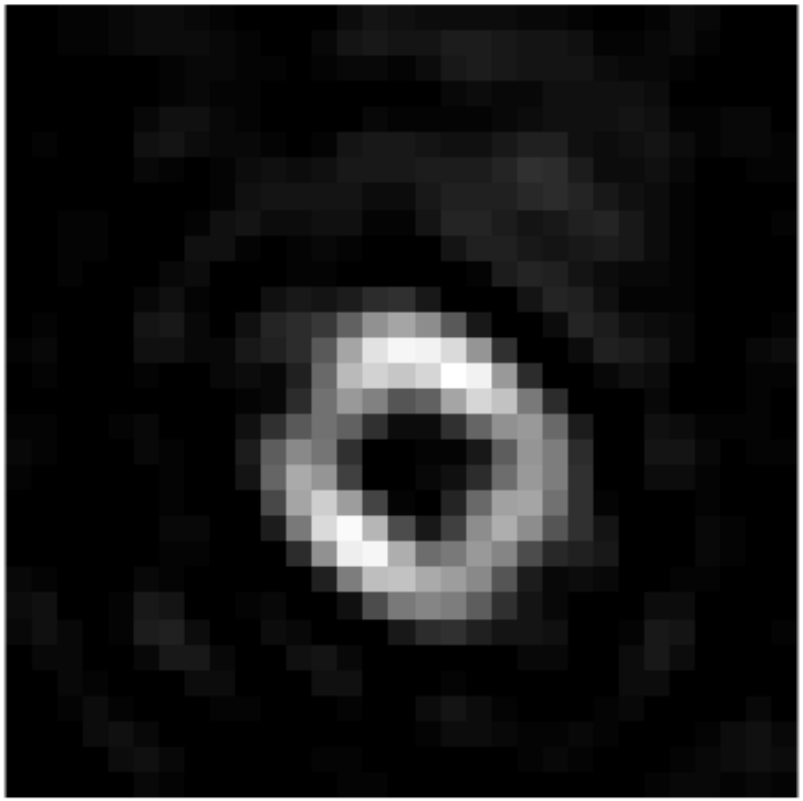


Figure: Kernel Estimation of patch 03

Numerical Result

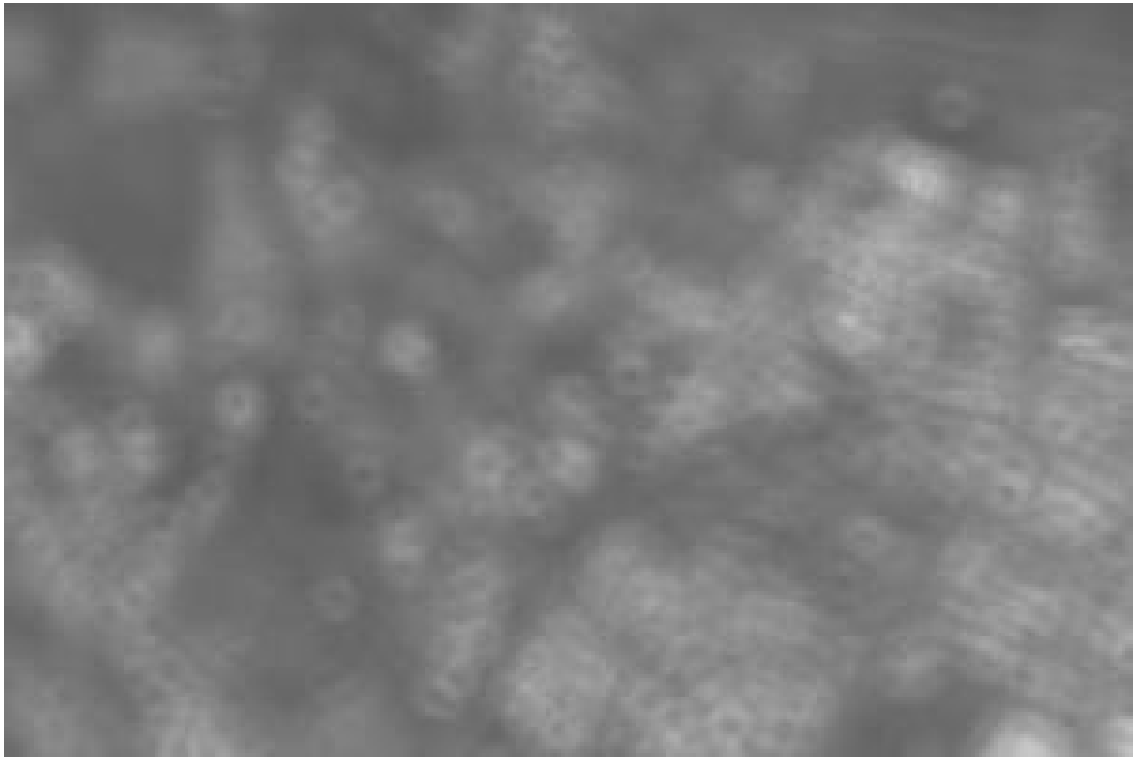


Figure: Patch 04

Numerical Result

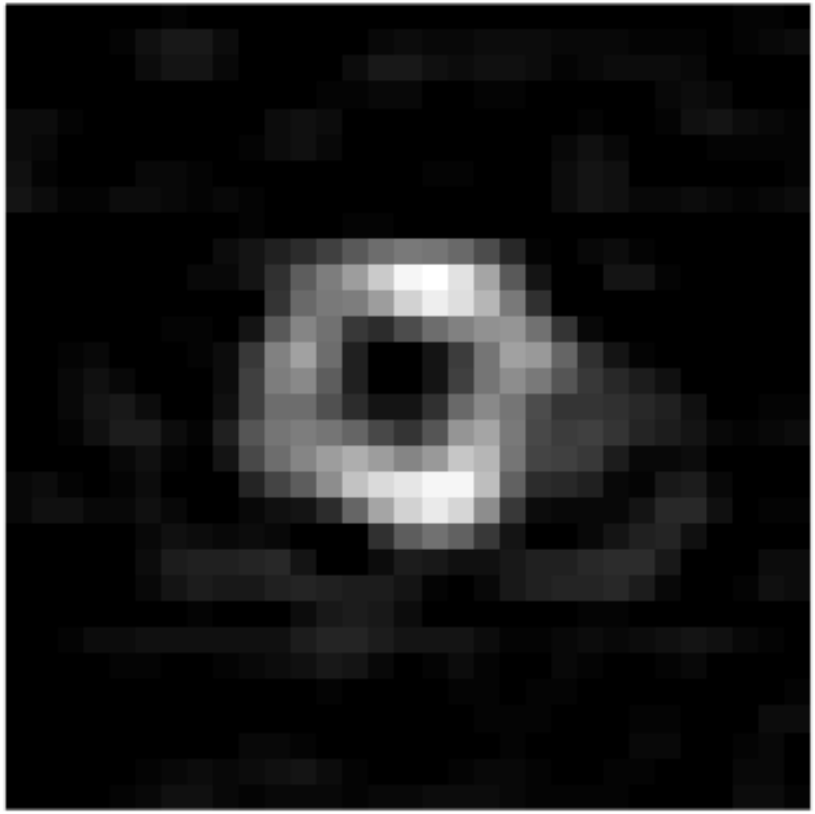


Figure: Kernel Estimation of patch 04

Q&A

Thank you for your listening!