Explore Image Deblurring via Blur Kernel Space

Phong Tran$^1$
Anh Tran$^{1,2}$
Quynh Phung$^1$
Minh Hoai$^{1,3}$

$^1$VinAI Research, Hanoi, Vietnam, $^2$VinUniversity, Hanoi, Vietnam, $^3$Stony Brook University, Stony Brook, NY 11790, USA

https://github.com/VinAIResearch/blur-kernel-space-exploring
Image Deblurring

Moving object

Camera shaking
Image Deblurring

Deblurring method
MAP-based Methods

\[ y = x \ast k + n \]

- **Y**: blur image
- **X**: sharp image
- **K**: blur kernel
- **N**: noise

Linear and uniform
MAP-based Methods

MAP Framework:

\[ x, k = \operatorname{argmax}_{x, k} \frac{P(y|x, k)P(x)}{P(k)} \]
MAP-based Methods

- Linear and uniform Gradient-based penalty, dark channels, ...
  
- Sparsity, Spectral properties, ...
MAP-based Methods

- Gradient-based penalty, dark channels, ...
- Does not hold in general
- Linear and uniform kernel
- Sparsity, Spectral properties, ...
- Linear and uniform
Deep Learning Models

CNN
Deep Learning Models - Challenges

Kernel overfitting

CNN
Our Work

- Generalize MAP-based method
- Leverage neural networks
Our Work

Assumptions:

\[ y = \mathcal{F}(x, k) \]

\[ \mathcal{F}(\cdot, k) \text{ : Blur operator parameterized by } k \]
Our Work

Assumptions:

\[ y = \mathcal{F}(x, k) \]

- \( \mathcal{F}(\cdot, k) \): Blur operator parameterized by \( k \)
- \( \mathcal{G}(x, y) \): Extract blur kernel \( k \) from \( (x, y) \)
Our Work

Find F and G
Our Work

Find F and G

Blind Deblurring
Our Work

Find F and G  Blind Deblurring  Blur Synthesis
Kernel Encoding

- F and G are implemented by two neural networks.
- For \((x, y) \sim P_{data}(x, y)\). F and G are jointly optimized by minimizing the objective function:

\[
\mathbb{E}_{x,y} \left[ \rho(y, F(x, G(x, y))) \right]
\]
Kernel Encoding

- F and G are implemented by two neural networks.
- For $(x, y) \sim P_{\text{data}}(x, y)$. F and G are jointly optimized by minimizing the objective function:

$$\mathbb{E}_{x,y} [\rho(y, \mathcal{F}(x, \mathcal{G}(x, y)))]$$

Charbonnier Loss
Generic Image Deblurring

- X and k are alternatively optimized by minimizing:

\[
\sum_{i=1}^{n} \rho(y_i, F(x_i, G(x_i, y_i)))
\]

Charbonnier Loss

Recon blurry image
Generic Image Deblurring

- X and k are alternatively optimized by minimizing:
  \[ \sum_{i=1}^{n} \rho(y_i, F(x_i, G(x_i, y_i))) \]

---

**Algorithm 1** Blind image deblurring

**Input:** blurry image \( y \)

**Output:** sharp image \( x \)

1: Sample \( z_x \sim \mathcal{N}(0, I) \)
2: Randomly initialize \( \theta_x \) of \( G^x_{\theta_x} \)
3: while \( \theta_x \) has not converged do
4: Sample \( z_k \sim \mathcal{N}(0, I) \)
5: Randomly initialize \( \theta_k \) of \( G^k_{\theta_k} \)
6: while \( \theta_k \) has not converged do
7: \( g_k \leftarrow \partial L(\theta_x, \theta_k) / \partial \theta_k \)
8: \( \theta_k \leftarrow \theta_k + \alpha \ast \text{ADAM}(\theta_k, g_k) \)
9: end while
10: \( g_x \leftarrow \partial L(\theta_x, \theta_k) / \partial \theta_x \)
11: \( \theta_x \leftarrow \theta_x + \alpha \ast \text{ADAM}(\theta_x, g_x) \)
12: end while
13: \( x = G^x_{\theta_x}(z_x) \)
Generic Image Deblurring

• X and k are alternatively optimized by minimizing:
\[ \sum_{i=1}^{n} \rho(y_i, F(x_i, G(x_i, y_i))) \]

Algorithm 1: Blind image deblurring

**Input:** blurry image y

**Output:** sharp image x

1: Sample \( z_x \sim \mathcal{N}(0, I) \)
2: Randomly initialize \( \theta_x \) of \( G^{x}_{\theta_x} \)
3: while \( \theta_x \) has not converged do
   4: Sample \( z_k \sim \mathcal{N}(0, I) \)
   5: Randomly initialize \( \theta_k \) of \( G^{k}_{\theta_k} \)
   6: while \( \theta_k \) has not converged do
      7: \[ g_k \leftarrow \frac{\partial \mathcal{L}(\theta_x, \theta_k)}{\partial \theta_k} \]
      8: \[ \theta_k \leftarrow \theta_k + \alpha \times \text{ADAM}(\theta_k, g_k) \]
   9: end while
10: \[ g_x \leftarrow \frac{\partial \mathcal{L}(\theta_x, \theta_k)}{\partial \theta_x} \]
11: \[ \theta_x \leftarrow \theta_x + \alpha \times \text{ADAM}(\theta_x, g_x) \]
12: end while
13: \( x = G_{\theta_x}(z_x) \)
Generic Image Deblurring

• X and k are alternatively optimized by minimizing:

\[
\sum_{i=1}^{n} \rho(y_i, F(x_i, G(x_i, y_i)))
\]

Algorithm 1 Blind image deblurring

**Input:** blurry image \(y\)

**Output:** sharp image \(x\)

1: Sample \(z_x \sim \mathcal{N}(0, I)\)
2: Randomly initialize \(\theta_x\) of \(G^{x}_{\theta_x}\)
3: while \(\theta_x\) has not converged do
4: Sample \(z_k \sim \mathcal{N}(0, I)\)
5: Randomly initialize \(\theta_k\) of \(G^{k}_{\theta_k}\)
6: while \(\theta_k\) has not converged do
7: \(g_k \leftarrow \frac{\partial \mathcal{L}(\theta_x, \theta_k)}{\partial \theta_k}\)
8: \(\theta_k \leftarrow \theta_k + \alpha \ast ADAM(\theta_k, g_k)\)
9: end while
10: \(g_x \leftarrow \frac{\partial \mathcal{L}(\theta_x, \theta_k)}{\partial \theta_x}\)
11: \(\theta_x \leftarrow \theta_x + \alpha \ast ADAM(\theta_x, g_x)\)
12: end while
13: \(x = G_{\theta_x}(z_x)\)
• X and k are alternatively optimized by minimizing:
\[ \rho(y, F(x, k)) + \lambda \| k \|_2 + \gamma (g_u^2(x) + g_v^2(x))^{\alpha/2} \]

\[ \text{Algorithm 1 Blind image deblurring} \]

\[ \text{Input:} \ \text{blurry image} \ y \]
\[ \text{Output:} \ \text{sharp image} \ x \]

1: Sample \( z_x \sim \mathcal{N}(0, I) \)
2: Randomly initialize \( \theta_x \) of \( G^{x}_{\theta_x} \)
3: while \( \theta_x \) has not converged do
4: \hspace{1em} Sample \( z_k \sim \mathcal{N}(0, I) \)
5: \hspace{1em} Randomly initialize \( \theta_k \) of \( G^{k}_{\theta_k} \)
6: \hspace{1em} while \( \theta_k \) has not converged do
7: \hspace{2em} \( g_k \leftarrow \partial \mathcal{L}(\theta_x, \theta_k) / \partial \theta_k \)
8: \hspace{2em} \( \theta_k \leftarrow \theta_k + \alpha \ast ADAM(\theta_k, g_k) \)
9: \hspace{1em} end while
10: \( g_x \leftarrow \partial \mathcal{L}(\theta_x, \theta_k) / \partial \theta_x \)
11: \( \theta_x \leftarrow \theta_x + \alpha \ast ADAM(\theta_x, g_x) \)
12: end while
13: \( x = G^{x}_{\theta_x}(z_x) \)
Generic Image Deblurring

- Deep Image Prior:
  - Replace $x$ by $G_{\theta_x}^y$
  - Replace $k$ by $G_{\theta_k}^k$

- $x$ and $k$ are alternatively optimized by minimizing:
  \[
  \rho(y, F(x, k)) + \lambda \|k\|_2 + \gamma(g_u^2(x) + g_v^2(x))^{\alpha/2}
  \]

Regularization term

---

Algorithm 1 Blind image deblurring

**Input:** blurry image $y$

**Output:** sharp image $x$

1: Sample $z_x \sim \mathcal{N}(0, I)$

2: Randomly initialize $\theta_x$ of $G_{\theta_x}^y$

3: while $\theta_x$ has not converged do

4: Sample $z_k \sim \mathcal{N}(0, I)$

5: Randomly initialize $\theta_k$ of $G_{\theta_k}^k$

6: while $\theta_k$ has not converged do

7: $g_k \leftarrow \partial \mathcal{L}(\theta_x, \theta_k)/\partial \theta_k$

8: $\theta_k \leftarrow \theta_k + \alpha * ADAM(\theta_k, g_k)$

9: end while

10: $g_x \leftarrow \partial \mathcal{L}(\theta_x, \theta_k)/\partial \theta_x$

11: $\theta_x \leftarrow \theta_x + \alpha * ADAM(\theta_x, g_x)$

12: end while

13: $x = G_{\theta_x}(z_x)$
Domain-specific Image Deblurring

\[ z^*, k^* = \arg \max_{z,k} \rho(\mathcal{F}(G_{style}(z), k), y) + R_z(z) + R_k(k) \]

Pre-trained StyleGAN
Blur Synthesis

\[ G(x_1, y_1) \]

\((x_1, y_1)\)
Blur Synthesis

\[ G(x_1, y_1) \]

\[ F(x_2, k_1) \]

\( (x_1, y_1) \)
Experimental Results – Kernel Encoding

Kernel 8

\( (x_1, y_1) \)

\( (x_2, y_2) \)
Experimental Results – Kernel Encoding

\[ G(x_1, y_1) \]

\[ F(x_2, k_1) \]

\[ x_2 \]

\[ y'_2 \]
Experimental Results – Kernel Encoding

$$G(x_1, y_1)$$

$$F(x_2, k_1)$$

PSNR

$$x_2$$

$$y'_2$$

$$y_2$$
## Experimental Results – Kernel Encoding

Blur transferring performance on Levin dataset

<table>
<thead>
<tr>
<th>PSNR (db)</th>
<th>kernel 1</th>
<th>kernel 2</th>
<th>kernel 3</th>
<th>kernel 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>49.48</td>
<td>51.93</td>
<td>52.06</td>
<td>53.74</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PSNR (db)</th>
<th>kernel 5</th>
<th>kernel 6</th>
<th>kernel 7</th>
<th>kernel 8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>49.91</td>
<td>49.49</td>
<td>51.43</td>
<td>50.38</td>
</tr>
</tbody>
</table>
## Experimental Results – Kernel Encoding

SRN performance when training on blur-swapped dataset

<table>
<thead>
<tr>
<th>Training data</th>
<th>Dataset</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>REDS</td>
<td>GOPRO</td>
</tr>
<tr>
<td>Original</td>
<td>30.70</td>
<td>30.20</td>
</tr>
<tr>
<td>Blur-swapped</td>
<td>29.43</td>
<td>28.49</td>
</tr>
</tbody>
</table>

SRN performance when training on blur-swapped dataset
Experimental Results – Generic Image Deblurring

Blur
SRN
Ours
Sharp
SelfDeblur
DeblurGANv2

–
Experimental Results – Generic Image Deblurring
Experimental Results – Blind Image Deblurring

Blur

SelfDeblur

DeblurGANv2
  imgaug

DeblurGANv2
  REDS

SRN imgaug

SRN REDS

Ours
## Experimental Results – Blind Image Deblurring

<table>
<thead>
<tr>
<th>Blur</th>
<th>SelfDeblur</th>
<th>DeblurGANv2</th>
<th>DeblurGANv2</th>
<th>SRN imaug</th>
<th>SRN REDS</th>
<th>ours</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="blur.png" alt="Image" /></td>
<td><img src="selfdeblur.png" alt="Image" /></td>
<td><img src="deblurGANv2.png" alt="Image" /></td>
<td><img src="deblurGANv2.png" alt="Image" /></td>
<td><img src="srn.png" alt="Image" /></td>
<td><img src="srn.png" alt="Image" /></td>
<td><img src="ours.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="blur.png" alt="Image" /></td>
<td><img src="selfdeblur.png" alt="Image" /></td>
<td><img src="deblurGANv2.png" alt="Image" /></td>
<td><img src="deblurGANv2.png" alt="Image" /></td>
<td><img src="srn.png" alt="Image" /></td>
<td><img src="srn.png" alt="Image" /></td>
<td><img src="ours.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="blur.png" alt="Image" /></td>
<td><img src="selfdeblur.png" alt="Image" /></td>
<td><img src="deblurGANv2.png" alt="Image" /></td>
<td><img src="deblurGANv2.png" alt="Image" /></td>
<td><img src="srn.png" alt="Image" /></td>
<td><img src="srn.png" alt="Image" /></td>
<td><img src="ours.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="blur.png" alt="Image" /></td>
<td><img src="selfdeblur.png" alt="Image" /></td>
<td><img src="deblurGANv2.png" alt="Image" /></td>
<td><img src="deblurGANv2.png" alt="Image" /></td>
<td><img src="srn.png" alt="Image" /></td>
<td><img src="srn.png" alt="Image" /></td>
<td><img src="ours.png" alt="Image" /></td>
</tr>
</tbody>
</table>
Experimental Results – Blur Synthesis

Source sharp

Source blur

Synthesized blur
Experimental Results – Blur Synthesis
Summary

- We have proposed a method to encode the blur kernel space of a deblurring dataset.
- We have proposed some applications of the blur kernel space.

Code

https://github.com/VinAIResearch/blur-kernel-space-exploring

Paper