Edge-based Blur Kernel Estimation Using Patch Priors
Supplementary Materials I
Details on Optimization Procedures

Libin Sun*
Brown University
lbsun@cs.brown.edu

Sunghyun Cho
Adobe Research
scho@adobe.com

Jue Wang
Adobe Research
juewang@adobe.com

James Hays
Brown University
hays@cs.brown.edu

In this document, we provide detailed mathematical derivations on the optimization procedures we have introduced in Sec. 4.1. in the paper. Specifically, we discuss the details on updating x and $\{\sigma^i\}$.

1. Derivations for Updating x

1.1. x-step Using $L2$ Penalty

To facilitate our discussion, we start out with a simpler objective function below, which is based on the L_2 norm instead of using the Lorentzian penalty on the patch similarity as done in the paper:

$$\begin{aligned} f_x(x) &= \sum_{\delta_*} \omega_* \|k * \delta_* x - \delta_* y\|^2 + \alpha \|\nabla x\|^2 \\ &+ \frac{\beta}{|M|} \sum_{i \in M} \|\overline{x^i} - \sigma^i Z^i\|^2 \\ &+ \gamma \sum_{i \in M} \left(\sigma^i - F_{ref}^{-1}(F_{\sigma,x}(\sigma^i))\right)^2 \end{aligned} \tag{1}$$

where all notations are as defined in the paper. Here the unknowns include $\{Z^i\}, \{\sigma^i\}, x$, and M is deterministically computed from x as discussed in the paper. The optimization requires that we sequentially update one of $\{Z^i\}$, $\{\sigma^i\}$, and x while holding the other two variables constant, until convergence. Given a current latent image x, updating $\{Z^i\}$ clearly boils down to a nearest neighbor search, which is trivial.

For easier derivation of the optimization process for x, we rewrite Eqn. (1) in a matrix form as:

$$\begin{aligned} f_{\mathbf{x}}(\mathbf{x}) &= \sum_{\delta_*} \omega_* \|\mathbf{K}\mathbf{D}_*\mathbf{x} - \mathbf{D}_*\mathbf{y}\|^2 \\ &+ \alpha \|\mathbf{D}_h \mathbf{x}\|^2 + \alpha \|\mathbf{D}_v \mathbf{x}\|^2 \\ &+ \frac{\beta}{|M|} \sum_{i \in M} \|\mathbf{P}_i \mathbf{x} - \mathbf{q}_i\|^2 \\ &+ \gamma \sum_{i \in M} \left(\sigma^i - F_{ref}^{-1}(F_{\sigma,x}(\sigma^i))\right)^2 \end{aligned} \tag{2}$$

where $\mathbf{K}$ and $\mathbf{D}_*$ represent the matrix form of k and δ_*, respectively, and $\mathbf{x}$ and $\mathbf{y}$ represent the vector form of x and y, respectively. $\mathbf{D}_h$ and $\mathbf{D}_v$ are the first order differentiation operators along the horizontal and vertical axes. $\mathbf{P}_i$ is a binary matrix extraction operator, extracting the patch at location i in the latent image x. $\mathbf{q}_i$ is defined as $\mathbf{q}_i = \sigma^i \mathbf{Z}^i + \boldsymbol{\mu}^i$ where $\mathbf{Z}^i$ is a vector representing Z^i.

We can update $\mathbf{x}$ by minimizing Eqn. (2) w.r.t. $\mathbf{x}$. This is equivalent to finding a solution of the following equation, which is derived by differentiating Eqn. (3) and setting it zero:

$$\left(\mathbf{A} + \frac{\beta}{|M|} \sum_{i \in M} \mathbf{P}_i^T \mathbf{P}_i\right) \mathbf{x} = \mathbf{b} + \frac{\beta}{|M|} \sum_{i \in M} \mathbf{P}_i^T \mathbf{q}_i \tag{3}$$

where $\mathbf{A}$ and $\mathbf{b}$ are defined as:

$$\mathbf{A} = \sum_{\delta_*} \omega_* \mathbf{K}^T \mathbf{D}_*^T \mathbf{D}_* \mathbf{K} + \alpha \sum_{\delta_x, \delta_y} \left(\mathbf{D}_h^T \mathbf{D}_h + \mathbf{D}_v^T \mathbf{D}_v\right),$$

$$\mathbf{b} = \sum_{\delta_*} \omega_* \mathbf{K}^T \mathbf{D}_*^T \mathbf{D}_* \mathbf{y}.$$

Since Eqn. (3) is a linear system w.r.t. $\mathbf{x}$, it can be solved efficiently using a biconjugate gradient descent method (`bicg` function in MATLAB), where the right-hand side is precomputed after fixing k, $\{Z^i\}$ and $\{\sigma^i\}$. Moreover, as $\mathbf{A}$ and $\mathbf{b}$ consist of convolution matrices, $\mathbf{A}\mathbf{x}$ and $\mathbf{b}$ can be computed efficiently using Fourier transforms as described in the paper.

*This research project was conducted while the first author was interning at Adobe Research.

1.2. x-step Using Lorentzian Penalty

The $L2$ penalty gives equal weights to all example patches Z^i, meaning all latent patches under the mask M will be optimized to be as similar to their respective example patches as possible. However, our patch priors are designed with restricted expressiveness, so there exist latent patches not captured in our example patch set. Such patches should be given smaller weights, so that x should be less affected by them.

One way of achieving this is through thresholding on the patch-pair distance and removing patches with large error values (above threshold) from the third term of Eqn. (1). However, it is not clear how to select an ideal threshold value. A better way of introducing such weighting scheme is through the use of robust statistics. A large family of robust penalty functions can be considered to alleviate this problem. Some popular ones include L_p norms with $0 \leq p < 2$, Huber loss function, Lorentzian function and Tukey's biweight function. For our framework, we choose the Lorentzian penalty function.

With the Lorentzian function, we rewrite our objective function in Eqn. (2) as:

$$\begin{aligned} f_{\mathbf{x}}(\mathbf{x}) &= \sum_{\delta_*} \omega_* \|\mathbf{K}\mathbf{D}_*\mathbf{x} - \mathbf{D}_*\mathbf{y}\|^2 \\ &+ \alpha\|\mathbf{D}_h\mathbf{x}\|^2 + \alpha\|\mathbf{D}_v\mathbf{x}\|^2 \\ &+ \frac{\beta}{|M|}\sum_{i\in M} \rho\left(\|\mathbf{P}_i\mathbf{x} - \mathbf{q}_i\|\right) \\ &+ \gamma \sum_{i\in M}\left(\sigma^i - F_{ref}^{-1}(F_{\sigma,x}(\sigma^i))\right)^2 \end{aligned} \quad (4)$$

where $\rho(r)$ is the Lorentzian function, defined as $\rho(r) = \log\left(1 + \frac{r^2}{2\epsilon^2}\right)$. Recall that $\mathbf{P}_i$ is the binary matrix operator extracting the ith patch from image $\mathbf{x}$. Based on the definition of the Lorentzian function, the third term on the right hand side of Eqn. (4) can be expanded as:

$$\rho(\|\mathbf{P}_i\mathbf{x} - \mathbf{q}_i\|) = \log\left(1 + \frac{1}{2\epsilon^2}\|\mathbf{P}_i\mathbf{x} - \mathbf{q}_i\|^2\right). \quad (5)$$

By differentiating Eqn. (5) w.r.t. $\mathbf{x}$, we obtain:

$$\frac{d\rho(\|\mathbf{P}_i\mathbf{x} - \mathbf{q}_i\|)}{d\mathbf{x}} = \frac{\mathbf{P}_i^T\mathbf{P}_i\mathbf{x} - \mathbf{P}_i^T\mathbf{q}_i}{\epsilon^2 + \frac{1}{2}\|\mathbf{P}_i\mathbf{x} - \mathbf{q}_i\|^2}. \quad (6)$$

By substituting Eqn. (6) into Eqn. (3), we get:

$$\left(\mathbf{A} + \frac{2\beta}{|M|}\sum_{i\in M} w_i\mathbf{P}_i^T\mathbf{P}_i\right)\mathbf{x} = \mathbf{b} + \frac{2\beta}{|M|}\sum_{i\in M} w_i\mathbf{P}_i^T\mathbf{q}_i \quad (7)$$

where $w_i = \left(2\epsilon^2 + \|\mathbf{P}_i\mathbf{x} - \mathbf{q}_i\|^2\right)^{-1}$. Note that Eqn. (7) is similar to Eqn. (3) *but* with weights w_i. This equation is no longer linear in $\mathbf{x}$, because w_i is a function of $\mathbf{x}$. Thus, we use an iterative reweighted least squares (IRLS) method to alternatingly optimize $\mathbf{x}$ and w_i.

2. Derivations for Updating $\{\sigma^i\}$

$\{\sigma^i\}$ is an important part of our formulation, designed specifically to restore image gradient variations destroyed by the blur process. Recall in Eqn. (4) only the last two terms involve $\{\sigma^i\}$, so we need to minimize the following objective function for updating $\{\sigma^i\}$, which is a subproblem in the x-step:

$$\begin{aligned} f_\sigma(\{\sigma^i\}) &= \frac{\beta}{|M|}\sum_{i\in M}\rho\left(\|\mathbf{P}_i\mathbf{x} - \sigma^i\mathbf{Z}^i - \boldsymbol{\mu}^i\|\right) \\ &+ \gamma\sum_{i\in M}\left(\sigma^i - \sigma_*^i\right)^2 \end{aligned} \quad (8)$$

where σ_*^i is defined as:

$$\sigma_*^i = F_{ref}^{-1}(F_{\sigma,x}(\sigma^i)). \quad (9)$$

Holding σ_*^i constant, we differentiate Eqn. (8) w.r.t. σ^i, then we have:

$$\begin{aligned} \frac{\partial f_\sigma(\{\sigma^i\})}{\partial\sigma^i} &= \frac{2\beta w_i}{|M|}\mathbf{Z}^{iT}\left(\mathbf{P}_i\mathbf{x} - \sigma^i\mathbf{Z}^i - \boldsymbol{\mu}^i\right) \\ &+ 2\gamma(\sigma^i - \sigma_*^i). \end{aligned} \quad (10)$$

Again, w_i is a function of σ^i, making direct optimization tricky. Thus, we adopt an IRLS and update $\{\sigma^i\}$ by iterating the following steps:

1. Compute $\{\sigma_*^i\}$ given current $\{\sigma^i\}$ using Eqn. (9).
2. Compute $\{w_i\}$ given current $\{\sigma^i\}$.
3. Update $\{\sigma^i\}$ by finding a solution which makes Eqn. (10) zero while holding $\{w_i\}$ constant, i.e:

$$\sigma^i \leftarrow \frac{w_i\beta/|M|\mathbf{Z}^{iT}\left(\mathbf{P}_i\mathbf{x} - \boldsymbol{\mu}^i\right) - \gamma\sigma_*^i}{w_i\beta/|M|\mathbf{Z}^{iT}\mathbf{Z}^i - \gamma} \quad (11)$$

4. Repeat the steps 2 and 3.

Note that Eqn. (9) can be efficiently solved by simple histogram matching.