Improving Image Restoration with Soft-Rounding: Supplementary Material

This document contains the following material:

1. Algorithm details on the \( x \) subproblem
2. Proof of Theorem 1
3. More visual examples on text image deconvolution, pattern image denoising and deconvolution.

1. The \( x \) Subproblem

Following the ALM framework in Sec. 3.2, the \( x \) subproblem (step 2) of our method is formulated as:

\[
\min_{x} \frac{1}{2} \|y - Kx\|^2_2 + \frac{\mu_I}{2} \|z^{k+1}_I + \frac{w_k}{\mu_I} - x\|^2_2 + \lambda_N \Gamma_N(Dx)
\]

where \( y \) is the degraded observation, \( K \) is the block Toeplitz matrix representation of the convolution kernel \( k \), \( z_I, w_I, \mu_I \) are the auxiliary variable, the Lagrange multiplier, and the penalty parameter for our distinct pixel value regularizer respectively, and \( D, \Gamma_N \) are the linear transform and the corresponding regularizer in the transformed domain. The first two terms are quadratic on \( x \), while the regularizer \( \Gamma_N \) is usually a non-smooth term on \( x \), which poses challenges for a direct numerical solution. For illustration purposes, we provide two \( \Gamma_N \) examples and their corresponding solutions for the \( x \) subproblem.

Example 1 We first consider the gradient-based \( \ell_0 \) regularizer [8, 11, 12], which has been used as the baseline method for text image/pattern image deconvolution in Sec. 4.1 and 4.2:

\[
\Gamma_N(\nabla x) = \|\nabla x\|_0
\]

where \( \nabla = (\nabla_h, \nabla_v) \) represents the gradient filters along horizontal and vertical directions, and the \( \ell_0 \) norm counts the number of nonzero elements in \( \nabla x \). Following the general variable splitting scheme [1], we introduce an auxiliary variable \( g \) and solve the equivalent problem, as:

\[
\min_{x} \frac{1}{2} \|y - Kx\|^2_2 + \frac{\mu_I}{2} \|v - x\|^2_2 + \lambda_N \|g\|_0
\]

where \( v = z^{k+1}_I + \frac{w_k}{\mu_I} \). We can solve this equality-constrained problem with half-quadratic splitting [3, 4] or the augmented Lagrangian method (ALM) [7]. For concise expression, we adopt the half-quadratic splitting technique and convert (3) to the following unconstrained problem:

\[
\min_{x, g} \frac{1}{2} \|y - Kx\|^2_2 + \frac{\mu_I}{2} \|v - x\|^2_2 + \lambda_N \|g\|_0
\]

When \( \beta \) approaches infinity, the solution of (4) reaches a local minimum of (3). Eq.(4) can be solved with block coordinate descent by alternatively minimizing \( x \) and \( g \) respectively. The \( x \) step is a least squares problem with three quadratic terms:

\[
\min_{x} \frac{1}{2} \|y - Kx\|^2_2 + \frac{\mu_I}{2} \|v - x\|^2_2 + \frac{\beta}{2} \|g - \nabla x\|^2_2
\]

which can be efficiently solved through FFT-based method [4]. The \( g \) step is formulated as:

\[
\min_{g} \frac{\beta}{2} \|g - \nabla x\|^2_2 + \lambda_N \|g\|_0
\]

which has a closed-form solution as follows [8, 11, 12]:

\[
g = \begin{cases} 
\nabla x, & |\nabla x|^2 \geq \frac{2\lambda_N}{\beta} \\
0, & \text{otherwise}.
\end{cases}
\]

Note that this solution can be easily adapted to solve other \( \Gamma_N \) regularizers such as total variation [10] (Sec. 4.3), generalized Laplacian [5] and EPLL [13]: we only need to change \( D \) and the updating equations of the \( g \) step according to the selected linear transform and the \( \Gamma_N \) regularizer.

Example 2 We then consider the denoising problem, where \( K \) reduces to an identity matrix and \( \Gamma_N \) represents a denoising regularizer such as NLDDL [9] and BM3D [2]. We can still introduce auxiliary variables and solve the \( x \) subproblem with half-quadratic splitting or ALM. However, for the special case of image denoising, this step can be much simplified. Since \( K \) is an identity matrix, the first two quadratic terms of (1) can be merged into one quadratic term as follows:

\[
\min_{x} \frac{1}{2} \|y - Kx\|^2_2 + \frac{\mu_I}{2} \|v - x\|^2_2 + \lambda_N \|g\|_0
\]
Eq.(8) is a standard Gaussian denoising problem, which can be solved with any existing denoisers. For the pattern denoising application (Sec. 4.2), we choose BM3D as our baseline method.

2. Proof of Theorem 1

For constant $\lambda > 0$ and $t_1 < t_2 \cdots < t_n$, we aim to find the solution to the following optimization problem

$$
\phi_{t, \lambda}(c) = \arg\min_x L(x) = \arg\min_x \left( \frac{1}{2\lambda}(x - c)^2 + \gamma_t(x) \right)
$$

where

$$
\gamma_t(x) = \begin{cases} 
\frac{1}{2}(t_1 - x) & \text{if } x < t_1 \\
\frac{1}{2}(x - t_j)(t_{j+1} - x) & \text{if } x \in [t_j, t_{j+1}] \\
\frac{1}{2}(x - t_n) & \text{if } x > t_n
\end{cases}
$$

(9)

**Theorem 1.** The optimal solution to (9) is given by:

(i) for $c \leq t_1$, $\phi_{t, \lambda}(c) = \min(t_1, c + \frac{\lambda}{2})$;

(ii) for $c \geq t_n$, $\phi_{t, \lambda}(c) = \max(t_n, c - \frac{\lambda}{2})$;

(iii) hard-rounding: for $c \in [t_1, t_n]$ and $\lambda \geq 1,$

$$
\phi_{t, \lambda}(c) = \begin{cases} 
t_j & c \in [t_j, t_j + d_j) \\
t_{j+1} & c \in [t_{j+1} - d_{j+1}, t_{j+1}]
\end{cases}
$$

(11)

where $d_j = \frac{1}{2}(t_{j+1} - t_j)$ and $j = 1, \cdots, n - 1$;

(iv) soft-rounding: for $c \in [t_1, t_n]$ and $\lambda \in (0, 1)$,

$$
\phi_{t, \lambda}(c) = \begin{cases} 
t_j & c \in [t_j, t_j + d_j) \\
t_{j+1} & c \in [t_{j+1} - d_{j+1}, t_{j+1}]
\end{cases}
$$

(12)

where $d_j = \frac{\lambda}{2}(t_{j+1} - t_j)$ and $j = 1, \cdots, n - 1$.

**Proof.** We first show that $c$ and $\phi_{t, \lambda}(c)$ always lie in the same interval. If $c \in [t_j, t_{j+1})$ (cases (iii) and (iv)), for any $x \leq t_j$, we have

$$
L(x) = \frac{1}{2\lambda}(x - c)^2 + \gamma_t(x) \geq \frac{1}{2\lambda}(t_j - c)^2 = L(t_j)
$$

Similarly, for any $x \geq t_{j+1}$, we have $L(x) \geq L(t_{j+1})$. Therefore the minimum of $L$ is obtained in $[t_j, t_{j+1})$, i.e., $\phi_{t, \lambda}(c) \in [t_j, t_{j+1})$. The same conclusion also holds for boundary cases (i) and (ii).

We then minimize $L(x)$ case by case. For case (i), we know $\phi_{t, \lambda}(c) \leq t_1$, which means we need to seek the minimum of $L(x)$ in $(-\infty, t_1)$:

$$
L(x) = \frac{1}{2\lambda}(x - c)^2 + \frac{1}{2}(t_1 - x), x \in (-\infty, t_1]
$$

The first derivative of $L(x)$ is:

$$
L'(x) = \frac{1}{\lambda}(x - c - \frac{1}{2}), x \in (-\infty, t_1]
$$

If $c + \frac{1}{2} \leq t_1$, $L(x)$ reaches the minimum when $L'(x) = 0$, which leads to $\phi_{t, \lambda}(x) = c + \frac{1}{2}$; if $c + \frac{1}{2} > t_1$, $L'(x) < 0$ for $x \leq t_1$, which means $L(x)$ reaches the minimum at $\phi_{t, \lambda}(c) = t_1$. Therefore, for case (i), $\phi_{t, \lambda}(c) = \min(c + \frac{1}{2}, t_1)$. Similarly, for case (ii), the optimal solution would be $\phi_{t, \lambda}(c) = \max(c - \frac{1}{2}, t_n)$.

If $c \in [t_j, t_{j+1})$, the minimum of $L(x)$ also locates in $[t_j, t_{j+1})$:

$$
L(x) = \frac{1}{2\lambda}(x - c)^2 + \frac{1}{2}(x - t_j)(t_{j+1} - x), x \in [t_j, t_{j+1}]
$$

We compute the first and second derivative of $L(x)$ as

$$
\begin{cases} 
L'(x) = \frac{1}{\lambda}x + \frac{1}{2}(t_j + t_{j+1}) - \frac{\lambda}{2} \\
L''(x) = \frac{1}{\lambda} + \frac{1}{2}
\end{cases}
$$

(11)

If $\lambda \geq 1$ (case (iii)), $L(x)$ does not have a minimum in the range of $[t_j, t_{j+1}]$ since $L''(x) \leq 0$, so the optimum can only occur on the boundary points of interval, i.e., $t_j$ or $t_{j+1}$, depending on which is closer to $c$. The optimal solution would be a rounding operation on $c$, as shown by (11).

If $0 < \lambda < 1$ (case (iv)), we find the point $x^*$ that satisfies $L'(x^*) = 0$:

$$
x^* = \frac{c}{1 - \lambda} - \frac{\lambda(t_j + t_{j+1})}{2(1 - \lambda)}
$$

If $x^* \in [t_j, t_{j+1})$, $L(x)$ reaches the minimum at $x^*$, i.e., $\phi_{t, \lambda}(c) = x^*$. The condition $t_j \leq x^* \leq t_{j+1}$ can then be reformulated as:

$$
t_j + \frac{1}{2}(t_{j+1} - t_j) \leq c \leq t_{j+1} - \frac{1}{2}(t_{j+1} - t_j)
$$

Therefore if $c$ satisfies the above condition, $\phi_{t, \lambda}(c) = x^*$; otherwise $\phi_{t, \lambda}(c)$ still takes the boundary values $t_j$ or $t_{j+1}$, depending on which is closer to $c$. The final form of the optimal solution is represented by Eq.(12).

3. More Visual Examples

Figure 1 shows the three motion-blur kernels we used for the deconvolution experiments in Sec.4. They have been used for performance evaluation in previous work [6, 8].

Figures 2, 3 and 4 show three text deconvolution examples with three kernel/noise settings as reported in the paper.

Figure 5 shows two pattern image denoising examples with different noise settings.

Figures 6, 7 and 8 show three L0 deconvolution examples of practical pattern images.

Visual details are best viewed on screen. We will provide more examples in an extended version.
Figure 1: Three motion-blur kernels we used for all the deconvolution experiments.

References


Figure 2: Text Image Deconvolution Example 1. The original Chinese text image is degraded with a $33 \times 33$ kernel and 3% Gaussian noise. The distinct pixel values for text and background pixels are $t_1 = 26$, $t_2 = 220$ respectively. Details are best viewed on screen.
Figure 3: Text Image Deconvolution Example 2. The hand-drawing graph image is degraded with a $45 \times 45$ kernel and 2% Gaussian noise. The distinct pixel values for foreground and background pixels are $t_1 = 39$, $t_2 = 220$ respectively. Details are best viewed on screen.
Figure 4: Text Image Deconvolution Example 3. The English text image is degraded with a $51 \times 51$ kernel and 1\% Gaussian noise. The distinct pixel values for foreground and background pixels are $t_1 = 0, t_2 = 255$ respectively. Details are best viewed on screen.
Figure 5: Pattern Image Denoising Examples. Top row: this pattern image is degraded with 25% Gaussian noise. The distinct pixel values are 0, 29, 76, 226, 255 respectively. Bottom row: this pattern image is degraded with 20% Gaussian noise. The distinct pixel values are 194 and 219 respectively. Details are best viewed on screen.

Figure 6: L0 Deconvolution Example 1. The 2D barcode image is degraded with a 101 × 101 kernel and 1% Gaussian noise. The distinct pixel values for barcode and background pixels are \( t_1 = 0, t_2 = 255 \) respectively. Details are best viewed on screen.
Figure 7: L0 Deconvolution Example 2. The movie poster image is degraded with a $65 \times 65$ kernel and 1% Gaussian noise. The distinct pixel values are 0, 71, 117, 149, 200 respectively. Details are best viewed on screen.

Figure 8: L0 Deconvolution Example 3. The word cloud image is degraded with a $55 \times 55$ kernel and 2% Gaussian noise. The distinct pixel values are 43, 56, 255 respectively. Details are best viewed on screen.