Fast Image/Video Upsampling
Supplementary File

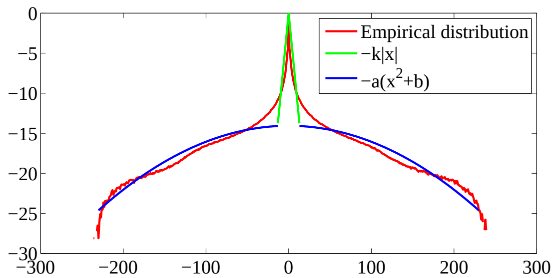

Figure 1: Logarithmic density of the image gradients (shown in red). Two piece-wise continuous functions (shown in green and blue) are employed to approximate it. The fitting parameters in this example are set as: $k = 1.0584$, $a = 2e^{-4}$, $b = 13.9651$, and $l_t = 13$.

1 The Deconvolution Process

Our deconvolution method is slightly modified from that in [Shan et al. 2008]. We illustrate one *logarithmic* image gradient density in Figure 1 calculated on a coarse-resolution image. It is used for imposing a regularization term in the deconvolution process. This heavy-tailed distribution is fitted with a piece-wise convex function

$$\Phi(x) \quad = \quad \begin{cases} -k|x| & x \le l_t \\ -(ax^2+b) & x > l_t \end{cases}, \tag{1}$$

where x denotes the image gradient level and l_t indexes the position where the two functions are concatenated. In Figure 1, $-k|x|$, shown in green, represents the sharp peak in the distribution at the center, while $-(ax^2+b)$ models the heavy tails of the distribution. $\Phi(x)$ is central-symmetric, and k, a, and b are the curve fitting parameters computed by minimizing the least square fitting error.

By incorporating such prior into the regularization term for the deconvolution process, we are subject to minimize the following energy function:

$$E(H) \propto \|f \otimes H - \tilde{H}\|_2^2 + \lambda_1(\|\Phi(\partial_x H)\|_1 + \|\Phi(\partial_y H)\|_1), \tag{2}$$

where $\partial_x H$ and $\partial_y H$ respectively denote the values of the x- and y-direction gradient, and λ_1 is a weight. To make the optimization efficient and robust, we adopt the variables substitution scheme similar to that used in [Shan et al. 2008] to estimate H.

Specifically, we first use variables $\mu = (\mu_x, \mu_y)$ to substitute $\partial H = (\partial_x H, \partial_y H)$, and add an additional term to measure the difference between ∂H and μ. So, Eq. (2) can be approximated by

$$\begin{aligned} E(H,\mu) \;\; = \;\; & \|f \otimes H - \tilde{H}\|_2^2 + \lambda_1(\|\Phi(\mu_x)\|_1 + \|\Phi(\mu_y)\|_1) + \\ & \lambda_2(\|\mu_x - \partial_x H\|\|_2^2 + \|\mu_y - \partial_y H\|\|_2^2), \end{aligned} \tag{3}$$

where λ_2 is a weight to control the relative importance that μ and ∂H have similar values. As described in [Shan et al. 2008], the purpose of using this variable substitution scheme is to separate $\Phi(\partial H)$ from $\|(f \otimes H) - \tilde{H}\|_2^2$ in optimization, thus making it possible to use FFT to accelerate the convolution process. The values of μ and ∂H will eventually be quite similar, since the weight λ_2 will be progressively increased in iterations. The following optimization is separated into two parts.

[μ step] In this step, we fix H to optimize μ. Eq. (3) becomes:

$$\begin{aligned} E(\mu) \;\; = \;\; & \lambda_1(\|\Phi(\mu_x)\|_1 + \|\Phi(\mu_y)\|_1) + \\ & \lambda_2(\|\mu_x - \partial_x H\|\|_2^2 + \|\mu_y - \partial_y H\|\|_2^2). \end{aligned} \tag{4}$$

Considering all pixels (x, y) in the image, $E(\mu)$ can be further written as $E(\mu) = \sum_{x,y}(E(\mu_x(x,y)) + E(\mu_y(x,y)))$. Since each $E(\mu(x,y))$ is not related to other energy elements, and it only contains one variable $\mu(x,y)$, we decompose the multi-variable optimization problem into a set of single-variable minimization problems. $E(\mu(x,y))$ consists of convex, differentiable pieces, each of which is minimized separately and the minimum among them is chosen. This optimization step can be completed quickly, resulting in a global minimum for $E(\mu(x,y))$.

[H step] In this step, we fix μ to minimize H. Eq. (3) becomes:

$$E(H) = \|f \otimes H - \tilde{H}\|_2^2 + \lambda_2(\|\mu_x - \partial_x H\|\|_2^2 + \|\mu_y - \partial_y H\|\|_2^2),$$

All the terms in the above function are in quadratic forms. We then apply Plancherel's theorem to the above derivation and get

$$\begin{aligned} & E(H) = \|\mathcal{F}(f) \circ \mathcal{F}(H) - \mathcal{F}(\tilde{H})\|_2^2 + \\ & \lambda_2\big(\|\mathcal{F}(\mu_x) - \mathcal{F}(\partial_x) \circ \mathcal{F}(H)\|_2^2 + \|\mathcal{F}(\mu_y) - \mathcal{F}(\partial_y) \circ \mathcal{F}(H)\|_2^2\big). \end{aligned}$$

By setting $\partial E(H)/\partial \mathcal{F}(H) = 0$, we obtain an optimal $\mathcal{F}^*(H)$ that minimize $E(H)$:

$$\mathcal{F}^*(H) = \frac{\overline{\mathcal{F}(f)} \circ \mathcal{F}(\tilde{H}) + \lambda_2 \overline{\mathcal{F}(\partial_x)} \circ \mathcal{F}(\mu_x) + \lambda_2 \overline{\mathcal{F}(\partial_y)} \circ \mathcal{F}(\mu_y)}{\overline{\mathcal{F}(f)} \circ \mathcal{F}(f) + \lambda_2 \overline{\mathcal{F}(\partial_x)} \circ \mathcal{F}(\partial_x) + \lambda_2 \overline{\mathcal{F}(\partial_y)} \circ \mathcal{F}(\partial_y)}.$$

Finally, the optimal solution can be computed by applying inverse Fourier Transform: $H^* = \mathcal{F}^{-1}(\mathcal{F}^*(H))$. The above two steps iterate until convergence. We set $\lambda_2 = 20$ initially. Then, we triple its value in each iteration to make μ and ∂H similar at convergence. λ_1 is adjustable in range $[0.01 - 0.3]$ in our experiments.

In the deconvolution step, most of the computation time is spent on computing FFTs in the H step, specifically, in each iteration, three FFTs are performed to compute $\mathcal{F}(\mu_x)$, $\mathcal{F}(\mu_y)$, and $\mathcal{F}^{-1}(\mathcal{F}^*(H))$ respectively. Outside the iterations, $\mathcal{F}(\tilde{H})$ only needs to be computed once and the result can be used for all iterations. Therefore, with n iterations, we need to perform a total of $3 * n + 1$ FFTs, where $n = 4$ in our implementation.

References

SHAN, Q., JIA, J. Y., AND AGARWALA, A. 2008. High-quality motion deblurring from a single image. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2008).*