Fast Image Deconvolution using Hyper-Laplacian Priors Supplementary Material

Dilip Krishnan,
Dept. of Computer Science,
Courant Institute,
New York University
dilip@cs.nyu.edu

Rob Fergus,
Dept. of Computer Science,
Courant Institute,
New York University
fergus@cs.nyu.edu

This document is supplementary material to our NIPS 2009 paper [1] of the same name. While we choose to solve the $\mathbf{w}$ sub-problem (Eqn. 5) in [1] using a LUT or analytically (for some specific values of α), a number of numerical alternatives exist. The simplest and fastest approach is to use Newton-Raphson (NR) to find the roots of the polynomials in Eqn. 10 and Eqn 11 of [1]. As shown in Table 1, 4 iterations of NR take a similar time to our analytic solution. However, the numerical algorithm has poor stability, particularly when the polynomials skim the x-axis. In practice, we found it gave spurious solutions around 1-2% of the time, an unacceptable rate given the iterative nature of our overall algorithm. Applying NR directly to Eqn. 5 of [1] is similarly unreliable. Although techniques with greater stability than NR exist, they are computationally more expensive. For example, we also compare to Matlab's `roots` command, based on solving an eigenvalue problem, but this was several orders of magnitude slower than our analytic approach.

Another benefit of our algorithms is that they are easily parallelizable. In addition to using Matlab's `.*` and `.^` operators, Algorithms 2 and 3 in [1] can be multi-threaded using the Intel Math Kernel Library (MKL) to give a $\sim$2 times speedup on a quad-core CPU (see Table 1). Lastly, the LUT approach can be used for values of α not restricted to $1/2$ or $2/3$.

Figure 1 shows the advantage of using a LUT (or analytic method) over using a Newton-Raphson method to solve the $\mathbf{w}$ sub-problem. Even though the SNR's of the output images are similar, we see the presence of spurious spikes where the NR root-finder could not find the true solution. These spikes do not contribute to the overall SNR score, but they degrade the visual quality of the output result. These errors could be fixed by using a safeguarded Newton's method, but that would increase the cost of finding the solution to the $\mathbf{w}$ sub-problem.

Polynomial	LUT w/MT	Analytic w/MT	Analytic	NR	`roots`
Cubic (α=1/2)	0.03	0.52	0.84	1.06	63
Quartic (α=2/3)	0.03	0.60	1.25	1.13	68

Table 1: Time (secs) to solve Eqn. 5 in [1] for a 1024x1024 image using different methods. Our analytic method and Newton-Raphson (NR) are comparable in speed. The LUT method is significantly faster. `roots` is Matlab's root finding algorithm. MT = multi-threading using Intel's MKL (1 CPU, 4 cores). For comparison, an `fft2` operation on the image with the same CPU takes 0.09 seconds.

References

[1] D. Krishnan and R. Fergus. Fast image deconvolution using hyper-laplacian priors. *To appear in proceedings of Neural Information Processing Systems*, 2009.

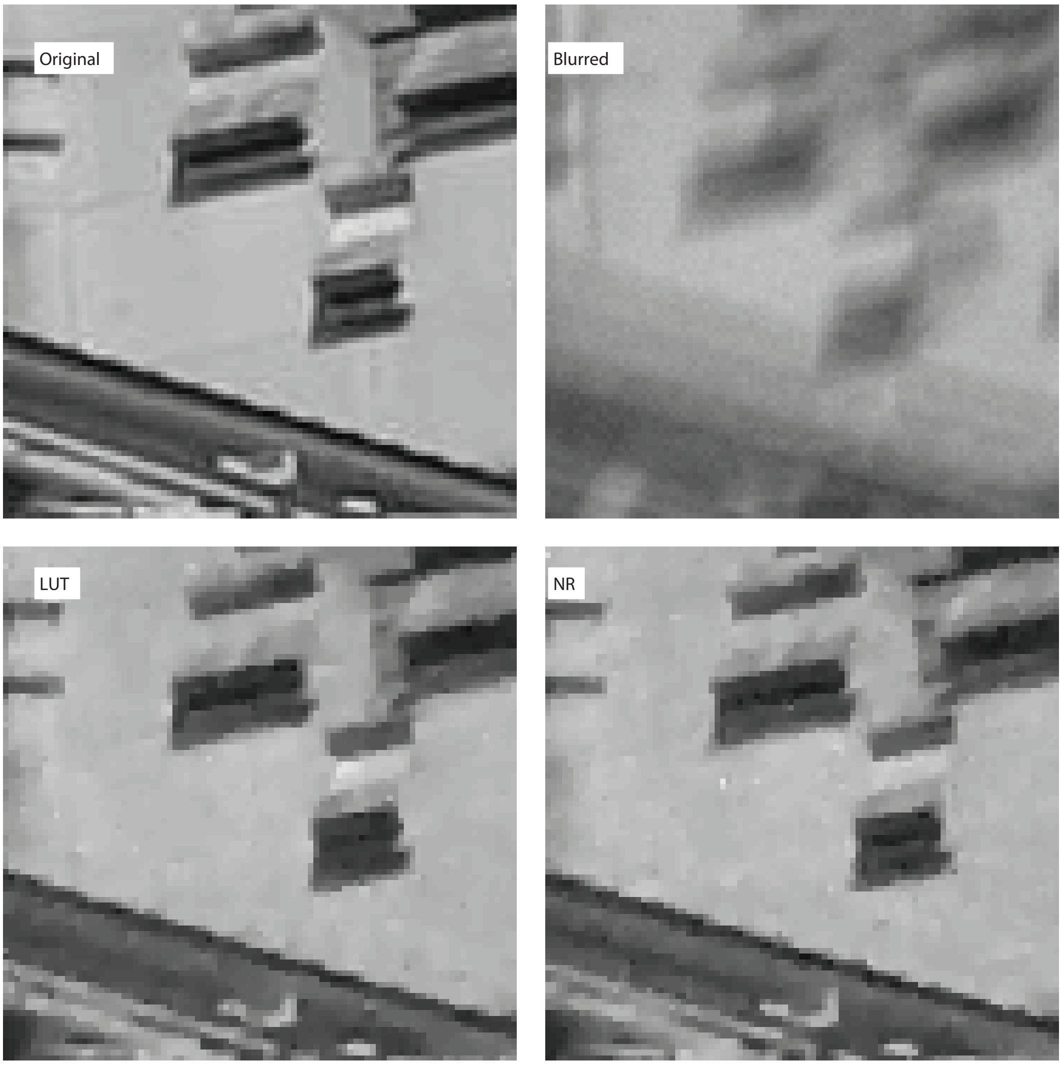

Figure 1: Crops from an image being deconvolved by the LUT and Newton-Raphson (NR) algorithms, using a 27×27 kernel. Clearly, the NR output contains many spurious spikes.