

## A 28.8mW Accelerator IC for Dark Channel Prior Based Blind Image Deblurring

Po-Shao Chen, Yen-Lung Chen, Yu-Chi Lee, Zih-Sing Fu,  
Chia-Hsiang Yang

National Taiwan University, Taipei, Taiwan

Image deblurring can be applied to mitigate the effect introduced by relative movement between object and camera, as shown in Fig. 1(a). Blind deblurring restores sharp images without knowing the camera motion, but it is also very time consuming. For example, it takes 11.4 minutes to process a full-HD image for [1]. Since the blur kernel estimation occupies 99% of blind deblurring process, a dedicated accelerator that utilizes the prior knowledge of gradients for real-time deblurring on mobile devices is presented in [2], which reduces the processing time from 11.4 minutes to 1.7 seconds.

The dark channel prior based deblurring algorithm achieves better deblurring performance, especially for images with small-scale objects and saturated pixels. According to the experiments on an in-house image dataset, the dark channel prior based algorithm [3] achieves an average peak signal-to-noise ratio (PSNR) of 29.35, outperforming the non-dark channel prior one [1] with an average PSNR of 27.18. However, the complexity of the dark channel prior based deblurring algorithms is also much higher. Experimented on the same platform, the time complexity of [3] is 6.5x higher than that of [1]. To address this issue, this work presents an energy-efficient accelerator for dark channel prior based blind image deblurring.

Fig. 1(b) shows the workflow of the dark channel prior based blind deblurring algorithm [3]. In the pre-processing stage, an input blurred image is first processed in the Gaussian image pyramid, which is composed of a Gaussian filter and a bilinear interpolator for image reshaping. Boundary wrapping is then applied by solving the 2D Laplace equation to find the patches to be wrapped around the blurred image. This reduces the ringing artifacts at the boundary of latent images. Initially, the blurred image is used to determine a coarse blur kernel. The newly calculated latent image is then used in the blur kernel estimation, and the updated blur kernel is applied to solve the new latent image. The details of the blur kernel are refined by updating the latent image and interim blur kernel alternately. The conjugate gradient (CG) method [4] is applied to blur kernel estimation without compute-intensive matrix inversion.

Fig. 2 shows the mathematical formulation of the dark channel prior based deblurring algorithm. To find the optimal solution, the regularization terms related to the gradient and the dark channel of the latent image are added for latent image estimation. Because L0-norm regularization is included for optimizing the latent image, the algorithm is divided into three sub-problems for  $u$ ,  $g$ , and  $l$ , which are related to the dark channel, the gradient, and the latent image, respectively. In the sub-problem for  $u$ , the dark channel is extracted from the image. The values of the dark channel derived from the background objects are removed to facilitate latent image estimation. Larger gradient values are preserved as specified in the sub-problem for  $g$ . In the sub-problem for  $l$ , 2D discrete Fourier transform (DFT) is extensively applied to estimate the latent image. After that, blur kernel estimation is then employed to both latent image's and blurred image's gradients to extract the blur kernel.

Fig. 3 shows the architecture of the proposed deblurring accelerator. A pre-processing unit is designed to pre-process the blurred image through a Gaussian image pyramid with boundary wrapping. To estimate the latent image, a dark channel re-assigner computes the dark channel successively. A blur kernel and a latent image estimator are used for alternating minimization based on sub-problems for  $u$ ,  $g$ , and  $l$ . A 1D fast Fourier transform (FFT) engine is designed to support 2D DFT for latent image estimation. A CG solver is deployed to estimate the blur kernels by performing the CG method.

Fig. 4(a) shows the 2D Laplace equation solver in the pre-processing unit for smooth boundary estimation. By performing the Jacobi method [5] over multiple processing elements, the latency is reduced by 81% compared to the direct-mapped implementation. In the dark channel re-assigner, an insertion-based sorting engine [6] is tailored to perform comparison in a parallel manner, as shown in Fig. 4(b). A

sorting element stores the location of a pixel from a row/column of a dark channel patch. When pixels are moved out of the patch, the data stored in the sorting elements are shifted to fill up the vacancy. For a patch with 25x25 pixels, a 25-stage sorting engine reduces the latency for image access by 96%. In the blur kernel estimator, the key 2D convolution unit in the CG solver employs five multiply-accumulate (MAC) operators in parallel, reducing the latency by 79%, as shown in Fig. 4(c). Floating-point (FP) arithmetic is utilized to cover a wide numeric range ( $10^{-11}$  to  $10^6$ ) with less hardware cost. Due to the alternating minimization between the latent image and the blur kernel, the SRAMs for latent image estimation can be reused in the CG solver to reduce the area by 92%. With the latency reduction from the proposed Laplace equation solver, the dark channel re-assigner and the CG solver, a latency reduction of 76% for a 257x257 blurred image is achieved.

The 1D FFT engine is proposed to support the images with boundary wrapping and images of gradients. Fig. 5(a) shows that the baseline FFT engine needs to additionally handle radices of 7, 9, 11, 13, and 181 derived from [3]. The associated image sizes are reshaped for FFT mapping for  $2^n$ ,  $2^n \times 3$ , and  $2^n \times 5$  points. The area of the FFT engine is reduced by 89% by leveraging the reshaping technique. The Winograd FFT is applied to implement radix-3 and radix-5 with only additions and constant multiplications. In the latent image estimator, the second-order gradient calculator is proposed to support image smoothing. Fig. 5(b) shows that data locality is exploited to reduce the memory access times by 57%.

Fig. 6(a) shows the deblurring performance of the prior art [2] and this work. Synthesized blurred images are generated for functional verification. The estimated blur kernels are sent to the same deconvolution software package for image reconstruction. The deblurring performance is measured by the structural similarity (SSIM) between the ground truth and the deblurred image. This work achieves an SSIM of 0.9131 and outperforms prior art [2], which implements the algorithm of [1], with an SSIM of 0.8561.

Fig. 6(b) shows the performance comparison with the state-of-the-art design [2]. Fabricated in a 40-nm CMOS technology, the core area of the proposed accelerator is 3.98 mm<sup>2</sup>. The chip dissipates 28.8 mW at a clock frequency of 65 MHz from a 0.65V supply. The proposed deblurring accelerator supports up to 257x257 blurred images and 49x49 blur kernels, while prior art [2] only supports 128x128 blurred images and 29x29 blur kernels. This work supports 4x larger image size and 2.9x larger blur kernel size with even 10% less area. It dissipates 2.1x less power given the same latency of 1.7s. The proposed chip achieves 2562x acceleration over CPU for full HD image deblurring, while [2] accelerates deblurring by a factor of 402x. This work achieves 4x [7.5x] higher normalized area [energy] efficiency, despite the better capability of image deblurring. Fig. 7 shows the micrograph and the chip summary.

### Acknowledgment

This work is supported by Ministry of Science and Technology (MOST) of Taiwan and Intelligent & Sustainable Medical Electronics Research Fund in NTU. The authors also thank Taiwan Semiconductor Research Institute (TSRI) for technical support on chip design and fabrication.

### References:

- [1] A. Levin *et al.*, "Efficient marginal likelihood optimization in blind deconvolution," *CVPR*, pp. 2657-2664, June 2011.
- [2] P. Raina *et al.*, "An energy-scalable accelerator for blind image deblurring," *IEEE JSSC*, vol. 52, no. 7, pp. 1849-1862, July 2017.
- [3] J. Pan *et al.*, "Blind image deblurring using dark channel prior," *CVPR*, pp. 1628-1636, June 2016.
- [4] S. Cho *et al.*, "Fast motion deblurring," *ACM Trans. Graph.*, vol. 28, no. 5, article 145, Dec. 2009.
- [5] P. Moin, *Fundamental of Engineering Numerical Analysis*, pp.141-142, Aug. 2010.
- [6] Y.-C. Wu *et al.*, "A 135-mW fully integrated data processor for next-generation sequencing," *ISSCC*, pp. 252-253, Feb. 2017.

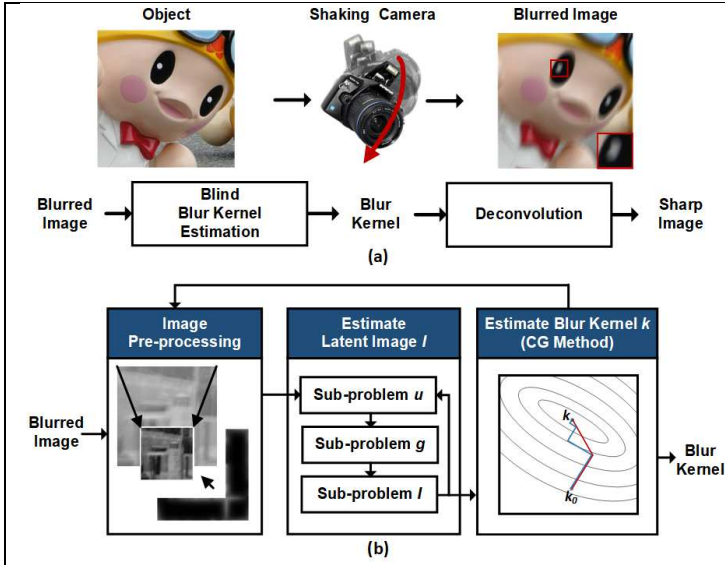


Fig. 1. Illustration of blind image deblurring and dark channel prior based blind deblurring algorithm.

Latent Image Estimation	Sub-problem $u$	$\min_{I, u, g} \ I * k - B\ _2^2 + \alpha \ \nabla I - g\ _2^2 + \beta \ D(I) - u\ _2^2 + \mu \ g\ _0 + \lambda \ u\ _0$
	Sub-problem $g$	$\min_u \beta \ D(I) - u\ _2^2 + \lambda \ u\ _0$
	Sub-problem $l$	$\min_g \alpha \ \nabla I - g\ _2^2 + \mu \ g\ _0$
	Sub-problem $l$	$I^{(i+1)} = \mathcal{F}^{-1} \left( \frac{\mathcal{F}(k) \mathcal{F}(I^{(i)}) + \beta (\mathcal{F}(\partial_x) \mathcal{F}(g_h) + \mathcal{F}(\partial_y) \mathcal{F}(g_v)) + \frac{\lambda}{\epsilon} \mathcal{F}(u)}{\mathcal{F}(k) \mathcal{F}(k) + \beta (\mathcal{F}(\partial_x) \mathcal{F}(\partial_x) + \mathcal{F}(\partial_y) \mathcal{F}(\partial_y)) + \frac{\lambda}{\epsilon}} \right)$
Blur Kernel Estimation		$\min_k \ Ak - b\ ^2 + \beta \ k\ ^2$

$B$ : blurred image  $I$ : latent image  $k$ : blur kernel  
 $D(I)$ : dark channel of the latent image  
 $\nabla I$ : gradient of the latent image  $u, g$ : auxiliary variables  
 $\alpha, \beta, \mu, \lambda$ : regularization terms  
 $\mathcal{F}$ : discrete Fourier transform (DFT)  
 $\partial_x, \partial_y$ : horizontal and vertical gradient operators  
 $g_h, g_v$ : horizontal and vertical gradient of the latent image  $I^{(i)}$   
 $\epsilon$ : threshold to differentiate background and foreground  
 $A$ : gradient of the latent image  $b$ : gradient of the blurred image  $k$ : blur kernel

Fig. 2. Formulation for the dark channel prior based deblurring algorithm.

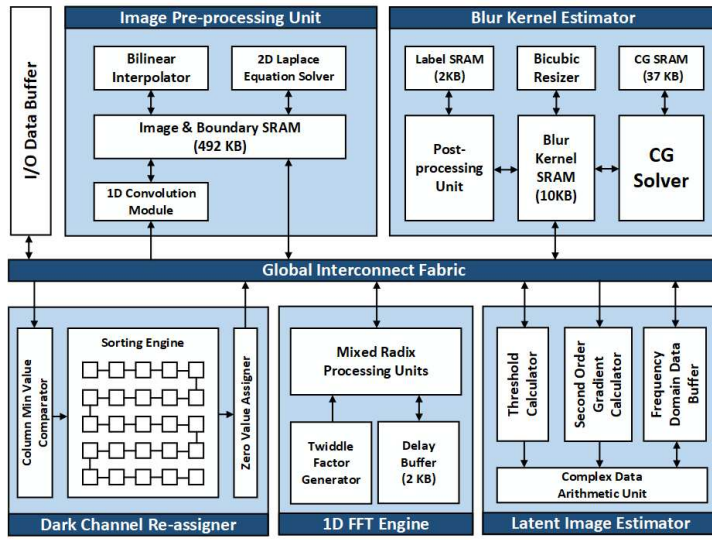


Fig. 3. System architecture of the proposed deblurring accelerator.

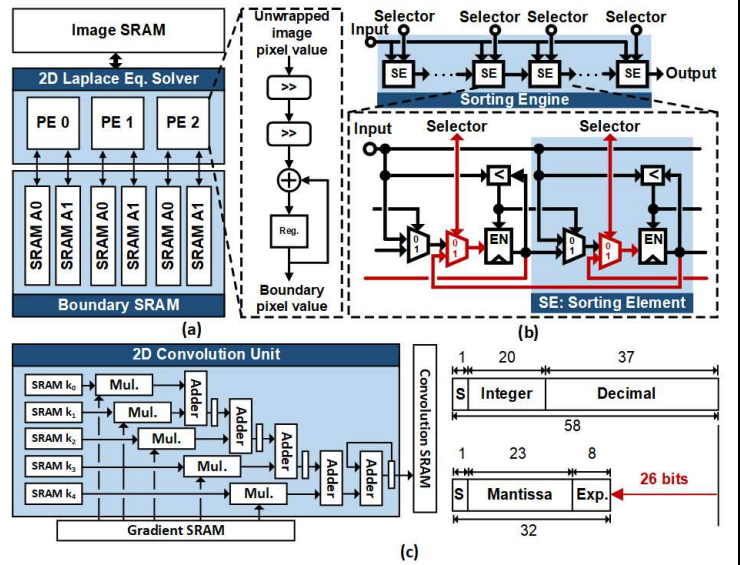


Fig. 4. (a) 2D Laplace equation solver, (b) sorting engine, (c) 2D convolution unit and arithmetic strength reduction in the CG solver.

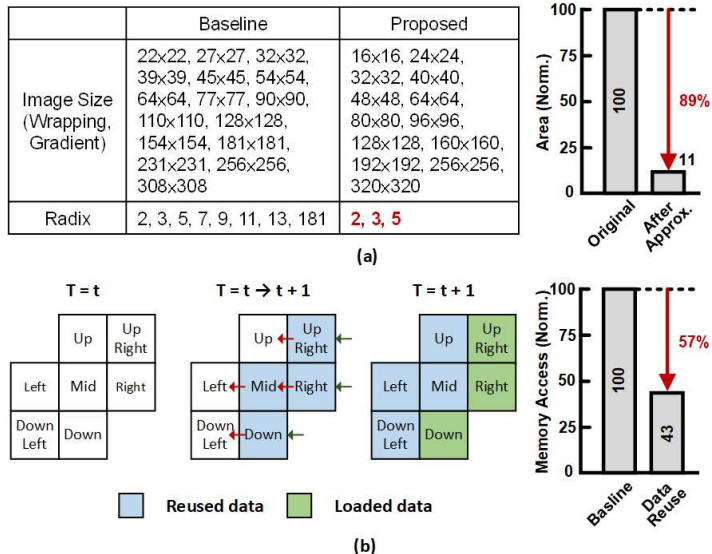


Fig. 5. (a) Radix reduction for the FFT engine and (b) memory access reduction for the second-order gradient calculator.

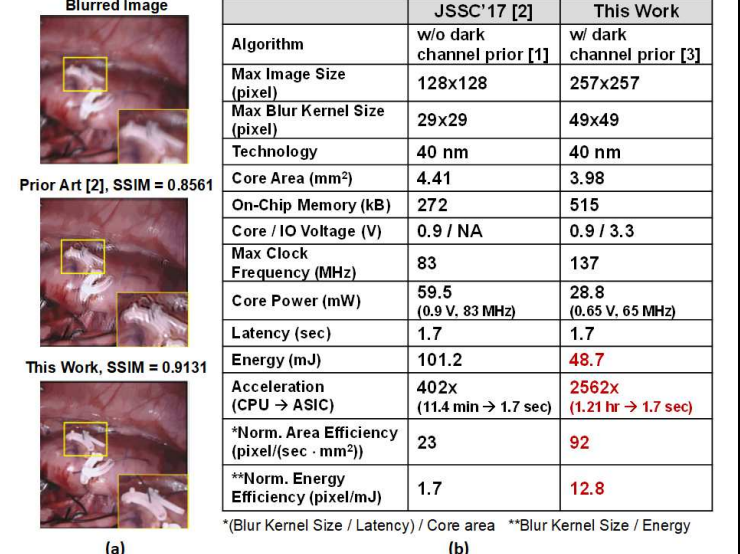


Fig. 6. (a) Deblurring performance and (b) comparison of deblurring accelerators.

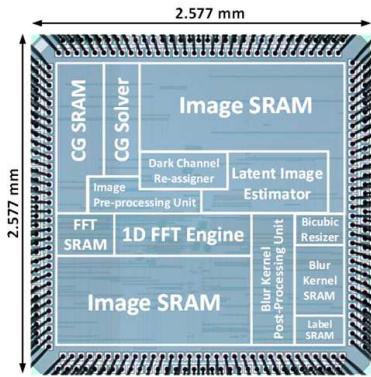
<div data-bbox="94 168 462 541"></div> <div data-bbox="475 235 792 495"><table><tr><td>Technology</td><td>40 nm CMOS</td></tr><tr><td>Supply Voltage (V)</td><td>0.6 – 0.9</td></tr><tr><td>Die Area (mm<sup>2</sup>)</td><td>2.577 × 2.577</td></tr><tr><td>Core Area (mm<sup>2</sup>)</td><td>1.995 × 1.995</td></tr><tr><td>On-chip Memory (kB)</td><td>515</td></tr><tr><td>Max Clock Frequency (MHz)</td><td>137</td></tr><tr><td>Image Size (pixel)</td><td>Up to 257 × 257</td></tr><tr><td>Blur Kernel Size (pixel)</td><td>Up to 49 × 49</td></tr><tr><td>Power (mW)</td><td>28.8 (@ 0.65 V, 65 MHz)</td></tr></table></div>	Technology	40 nm CMOS	Supply Voltage (V)	0.6 – 0.9	Die Area (mm <sup>2</sup> )	2.577 × 2.577	Core Area (mm <sup>2</sup> )	1.995 × 1.995	On-chip Memory (kB)	515	Max Clock Frequency (MHz)	137	Image Size (pixel)	Up to 257 × 257	Blur Kernel Size (pixel)	Up to 49 × 49	Power (mW)	28.8 (@ 0.65 V, 65 MHz)	
Technology	40 nm CMOS																		
Supply Voltage (V)	0.6 – 0.9																		
Die Area (mm <sup>2</sup> )	2.577 × 2.577																		
Core Area (mm <sup>2</sup> )	1.995 × 1.995																		
On-chip Memory (kB)	515																		
Max Clock Frequency (MHz)	137																		
Image Size (pixel)	Up to 257 × 257																		
Blur Kernel Size (pixel)	Up to 49 × 49																		
Power (mW)	28.8 (@ 0.65 V, 65 MHz)																		

Fig. 7. Chip micrograph and summary.