FAST IMAGE FILTERING WITH L^1 GAUSS TRANSFORM

Dina Bashkirova, Shin Yoshizawa, Hideo Yokota – **RIKEN**
{dina.bashkirova, shin, hyokota}@riken.jp

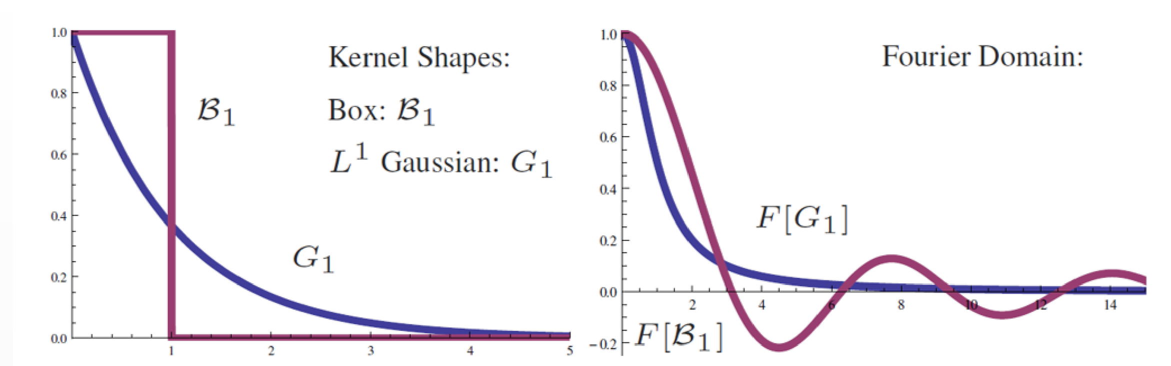
RIKEN's Programs for Junior Scientists

RIKEN

Abstract

Gaussian convolution is a core tool in many applications in computer vision and image processing. Since computing the exact Gaussian convolution on large datasets is a highly expensive task, a fast and accurate approximation technique is needed. We propose a novel 2D approximation approach based on L^1 distance metric and domain splitting that allows to significantly speed up the computational process compared to conventional methods while preserving high accuracy.

Why L^1 Kernel



L^1 Gauss Transform

L^1 distance between $\mathbf{x}_j = (x_j, y_j)$ and $\mathbf{x}_i = (x_i, y_i)$:

$$|\mathbf{x}_j - \mathbf{x}_i| = |x_j - x_i| + |y_j - y_i|$$

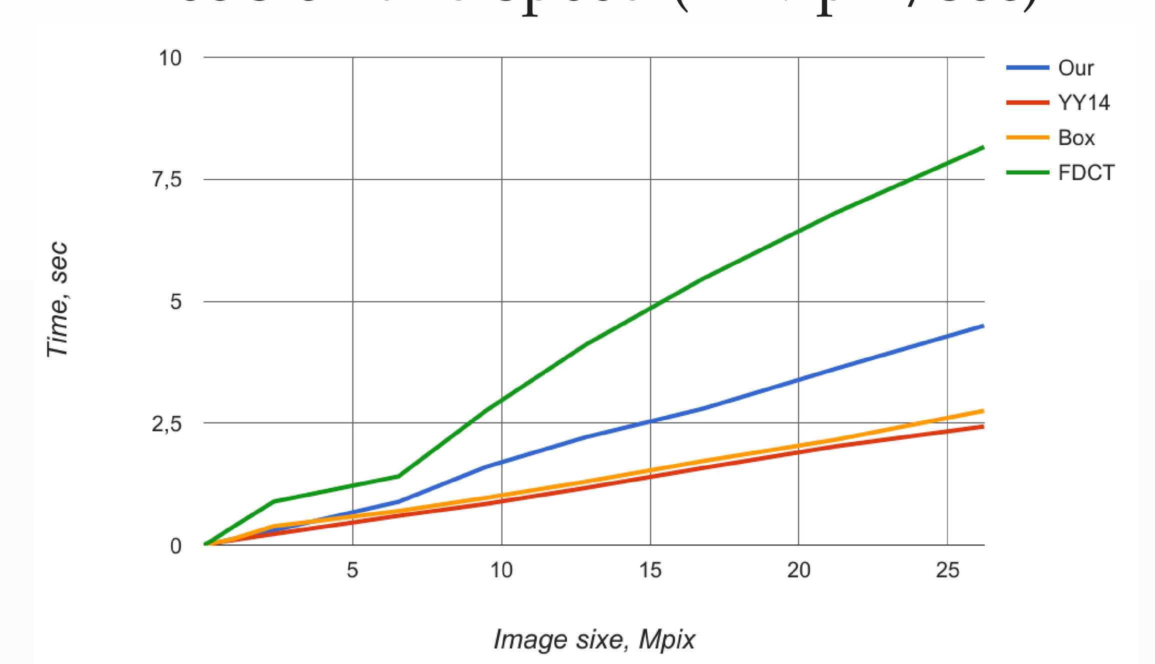
Gauss transform: $O(N^2)$ complexity

$$J(x_j) = \sum_{i=1}^{N} \exp\left(-\frac{|\mathbf{x}_j - \mathbf{x}_i|}{\sigma}\right) I_i$$

Experiments

	Our	YY14[1]	FDCT [4]	Box [3]
$E_{\max}$	$\mathbf{1.8 \times 10^{-11}}$	3.8×10^{-10}	0.44	3.73
PSNR	**291.05**	281.81	58.98	41.45
Speed	7.19	**9.76**	3.37	8.58

Precision and speed (in Mpix/sec).



This work was supported in part by Grants-in-Aid for Scientific Research of Japan (15H05954 and 16K15019).

Domain Splitting

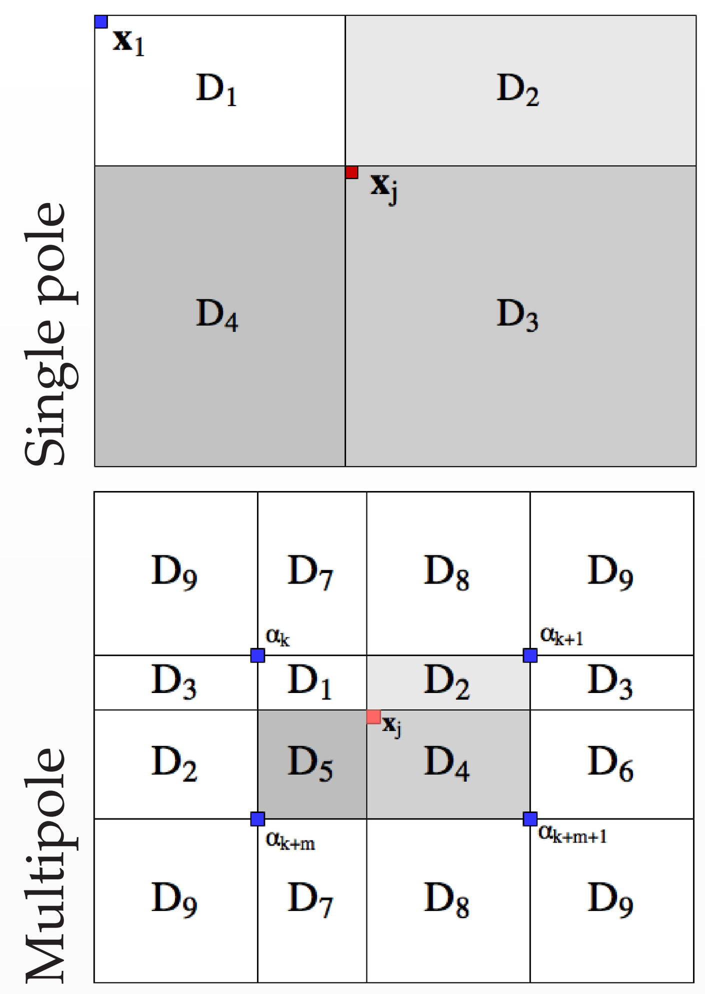


Image regions for point $\mathbf{x}_j$

$$|\mathbf{x}_j - \mathbf{x}_i| = \begin{cases} |x_j - x_1| - |x_i - x_1| + |y_j - y_1| - |y_i - y_1| & \text{if } _i \in D_1 \\ |x_i - x_1| - |x_j - x_1| + |y_j - y_1| - |y_i - y_1| & \text{if } _i \in D_2 \\ |x_j - x_1| - |x_i - x_1| + |y_i - y_1| - |y_j - y_1| & \text{if } _i \in D_3 \\ |x_i - x_1| - |x_j - x_1| + |y_i - y_1| - |y_j - y_1| & \text{if } _i \in D_4, \end{cases}$$

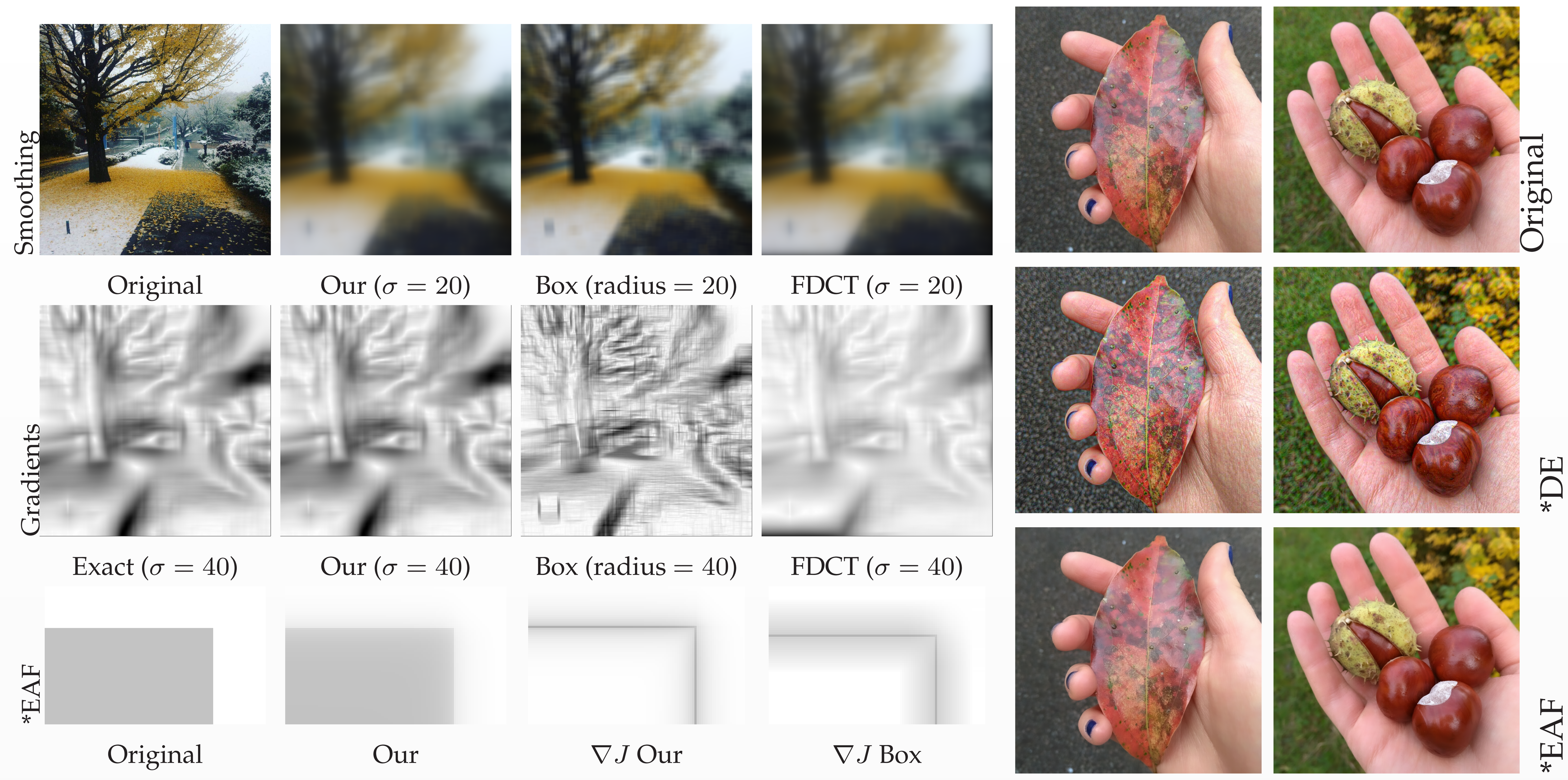
$$J(\mathbf{x}_j) = I(\mathbf{x}_j) + F(x_j)F(y_j) \sum_{\mathbf{x}_i \in D_1(j)} \frac{1}{F(x_i)F(y_i)} I(\mathbf{x}_i) +$$

$$+ \frac{F(x_j)}{F(y_j)} \sum_{\mathbf{x}_i \in D_2(j)} \frac{F(y_i)}{F(x_i)} I(\mathbf{x}_i) + \frac{F(y_j)}{F(x_j)} \sum_{i \in D_3(j)} \frac{F(x_i)}{F(y_i)} I(\mathbf{x}_i) +$$

$$+ \frac{1}{F(x_j)F(y_j)} \sum_{\mathbf{x}_i \in D_4(j)} F(x_i)F(y_i)I(\mathbf{x}_i),$$

where $F(x_j) \equiv G(x_j - x_1)$ and $F(y_j) \equiv G(y_j - y_1)$.

Filtering Comparisons



*EAF – Edge-Aware Filtering, *DE – Detail Enhancement.

Guided Filtering

$$K(\alpha, \beta) = \sum_{\mathbf{y} \in \Omega(\mathbf{x})} G(\mathbf{x} - \mathbf{y})(\alpha g(\mathbf{y}) + \beta - I(\mathbf{y}))^2 + \epsilon\alpha^2)$$

$$\alpha = \frac{f(Ig) - f(I)f(g)}{f(g^2) - f(g)^2 + \epsilon}, \quad \beta = f(I) - \alpha f(g)$$

$J(x) = f(\alpha)g(x) + f(\beta)$, $f(*)$ – smoothing filter

Future work

- Non-uniform case
- ND extension
- Apply to machine learning tasks: regression, segmentation, registration etc.

Contribution

+ Linear complexity ($O(N)$)
+ High precision
+ Improves Guided Filtering: quality and speed

References

[1] Yoshizawa, Shin et al., Fast L 1 Gaussian convolution via domain splitting, in *Image Processing (ICIP), IEEE IC*, 2014.

[2] He, Kaiming et al., Guided image filtering, in *IEEE TPAMI 35.6*, 2013.

[3] E. Dougherty, Digital Image Processing Methods, in *CRC Press*, 1994.

[4] T. Ooura, General Purpose FFT Package, in `http://www.kurims.kyoto-u.ac.jp/~ooura/fft.html`, 2006.