

Tiling Parallelization of Guided Image Filtering

Teppei Tsubokawa*, Masahiro Nakamura*[†], Yoshihiro Maeda*, and Norishige Fukushima*

* Nagoya Institute of Technology, Nagoya, Japan

Email: *fukushima@nitech.ac.jp

[†] Hitachi Solutions Create, Ltd.

Email: [†]asyuno.kakfow@gmail.com

Abstract—Guided image filtering is one of the edge-preserving filters. The filtering requires multiple times of box filtering to obtain average and variance/covariance values. We accelerate the guided image filtering by using fast box filtering. Also, we accelerate the filter by tiling for improving cache efficiency.

Index Terms—box filtering, guided filtering, cache efficiency, acceleration

I. INTRODUCTION

Box filtering is one of the smoothing filters, which compute averaging values in a kernel. The box filtering is often used in various applications for image processing and computer vision. One of the application examples of box filtering is guided image filtering [1].

The guided image filtering is one of the efficient edge-preserving filters. The filter is effective for blur or gradient regions thanks to the local linearity of this filtering. This filtering consists of multiple times of box filters, which are used for computing average and variance values in the local window.

In this paper, we use fast box filtering, which has high cache efficiency for multiple times of filters, for acceleration of the guided image filtering. Also, we divide the input image and guide image into tiles to improve cache efficiency.

II. RELATED WORKS

A. Efficient Box Filtering

1) *Box Filtering*: Box filtering is a simple linear filter with a square kernel and computes the average values in the kernel. Also, the box filtering is a 2D finite impulse response (FIR) filter. The computational order is $O(r^2)$, where r is the kernel radius. Computation of these filters has some practical solutions. A traditional approach is separable implementation. The 2D filter is decomposed into a combination of 1D FIR filters, and its computational order is $O(r)$. The definition of separable box filtering is as follows:

$$J(x, y) = \sum_{i=-r}^r I(x+i, y), \quad O(x, y) = \sum_{j=-r}^r J(x, y+j), \quad (1)$$

where I and O are input and output images, J is an intermediate image. O needs to be normalized by the size of kernel.

This work was supported by JSPS KAKENHI (JP17H01764, JP18K19813).

2) *Separable Summed-Area-Table*: Separable summed-area-table (SSAT) [2] is $O(1)$ order and requires two stages. The stages include horizontal and vertical filters. SSAT is defined as follows:

$$J(x, y) = J(x-1, y) + I(x+r, y) - I(x-r-1, y), \quad (2)$$

$$O(x, y) = O(x, y-1) + J(x, y+r) - J(x, y-r-1). \quad (3)$$

The filtering consists two additions, two subtractions and one normalization for each pixel.

3) *One-Pass Summed-Area-Table*: One-Pass Summed-Area-Table (OP-SAT) [3] is a method to convert the SSAT calculation into one stage. The OP-SAT is defined as follows:

$$J(x+r, y) = J(x+r, y-1) + I(x+r, y+r) - I(x+r, y-r-1), \quad (4)$$

$$O(x, y) = O(x-1, y) + J(x+r, y) - J(x-r-1, y). \quad (5)$$

Like the SSAT, OP-SAT needs only two additions, two subtractions and one normalization processing per one pixel. The implementation has high cache efficiency, since the calculation stage of OP-SAT is one stage.

B. Guided Image Filtering

Guided image filtering [1] is one of the efficient edge-preserving filters. We assume that output image q is a linear transform of the guidance image J . When the guidance image is grayscale, the output image is computed as follows:

$$q_i = \bar{a}_i J_i + \bar{b}_i, \quad (6)$$

$$\bar{a}_i = \frac{1}{|\omega|} \sum_{\mathbf{k} \in \omega_i} a_{\mathbf{k}}, \quad \bar{b}_i = \frac{1}{|\omega|} \sum_{\mathbf{k} \in \omega_i} b_{\mathbf{k}}, \quad (7)$$

where i and \mathbf{k} are pixel positions. ω_i is a set of neighborhood pixels around the pixel i . $|\omega|$ is the number of pixels of ω_i . $a_{\mathbf{k}}$ and $b_{\mathbf{k}}$ are linear transform coefficients, which are defined in Eq. (8) and (9), respectively. The linear coefficients are computed as follows:

$$a_{\mathbf{k}} = \frac{\frac{1}{|\omega|} \sum_{i \in \omega_{\mathbf{k}}} J_i p_i - \mu_{\mathbf{k}} \bar{p}_{\mathbf{k}}}{\sigma_{\mathbf{k}}^2 + \epsilon}, \quad (8)$$

$$b_{\mathbf{k}} = \bar{p}_{\mathbf{k}} - a_{\mathbf{k}} \mu_{\mathbf{k}}, \quad (9)$$

where $\mu_{\mathbf{k}}$ and $\sigma_{\mathbf{k}}^2$ are the average and variance of J in $\omega_{\mathbf{k}}$. \bar{p} is the average of the input image p in $\omega_{\mathbf{k}}$. ϵ is a regularization parameter.

Algorithm 1 Conventional implementation

- 1: $\text{average}_J = f_{\text{box}}(J, r)$
 $\text{average}_p = f_{\text{box}}(p, r)$
 $\text{corr}_J = f_{\text{box}}(J * J, r)$
 $\text{corr}_{Jp} = f_{\text{box}}(J * p, r)$
 - 2: $\text{var}_J = \text{corr}_J - \text{average}_J * \text{average}_J$
 $\text{cov}_{Jp} = \text{corr}_{Jp} - \text{average}_J * \text{average}_p$
 - 3: $a = \text{cov}_{Jp} / (\text{var}_J + \epsilon)$
 $b = \text{average}_p - a * \text{average}_J$
 - 4: $\text{average}_a = f_{\text{box}}(a, r)$
 $\text{average}_b = f_{\text{box}}(b, r)$
 - 5: $q = \text{average}_a * J + \text{average}_b$
-

Algorithm 2 Proposed implementation using OP-SAT

- 1: $\text{average}_J = f_{\text{OP-SAT}}(J, r)$
 $\text{average}_p = f_{\text{OP-SAT}}(p, r)$
 $\text{corr}_J = f_{\text{OP-SAT}}(J * J, r)$
 $\text{corr}_{Jp} = f_{\text{OP-SAT}}(J * p, r)$
 $\text{var}_J = \text{corr}_J - \text{average}_J * \text{average}_J$
 $\text{cov}_{Jp} = \text{corr}_{Jp} - \text{average}_J * \text{average}_p$
 $a = \text{cov}_{Jp} / (\text{var}_J + \epsilon)$
 $b = \text{average}_p - a * \text{average}_J$
 - 2: $\text{average}_a = f_{\text{OP-SAT}}(a, r)$
 $\text{average}_b = f_{\text{OP-SAT}}(b, r)$
 $q = \text{average}_a * J + \text{average}_b$
-

When the guidance image is color, a color guidance image can better preserve the edges, which are not distinguishable in grayscale. The output image is computed as follows:

$$q_i = \bar{\mathbf{a}}_i^T \mathbf{J}_i + \bar{b}_i, \quad (10)$$

$$\bar{\mathbf{a}}_i = \frac{1}{|\omega|} \sum_{\mathbf{k} \in \omega_i} \mathbf{a}_k, \quad \bar{b}_i = \frac{1}{|\omega|} \sum_{\mathbf{k} \in \omega_i} b_k, \quad (11)$$

where \mathbf{J}_i and \mathbf{a}_k represent 3×1 vectors. \mathbf{a}_k and b_k are linear coefficients, which is defined in Eq. (12) and (13), respectively. The linear coefficients are computed as follows:

$$\mathbf{a}_k = (\Sigma_k + \epsilon \mathbf{U})^{-1} \left(\frac{1}{|\omega|} \sum_{i \in \omega_k} \mathbf{J}_i p_i - \mu_k \bar{p}_k \right), \quad (12)$$

$$b_k = \bar{p}_k - \mathbf{a}_k^T \mu_k, \quad (13)$$

where Σ_k is the 3×3 covariance matrix of \mathbf{J} in ω_k , and \mathbf{U} is a 3×3 identity matrix.

III. PROPOSED METHOD

Algorithm 1 shows the conventional implementation of the guided image filtering, where f_{box} is a box filtering with a kernel radius r . The abbreviations in the Algorithm 1 are correlation (corr), variance (var), and covariance (cov), respectively. The conventional implementation needs five stages to obtain the output image.

As an Algorithm 2 shows the implementation of the guided image filtering using OP-SAT, which requires only two stages. In the proposed implementation, we calculate the average of

TABLE I: Computational times of guided image filtering [ms].

Image Size	512 × 512	512 × 768	1980 × 1088
SSAT without tiling	14.01	21.69	136.25
OP-SAT without tiling	10.96	16.69	94.81
SSAT with tiling	0.61 (32 × 4)	0.88 (16 × 8)	4.60 (64 × 16)
OP-SAT with tiling	0.81 (16 × 8)	1.08 (16 × 8)	5.60 (16 × 4)
Conventional	21.07	30.90	160.22

p, J, J^2, Jp at the same time at stage 1. Therefore, we have all the information, which is necessary for the calculation at stage 2 in Algorithm 1. Then, we can calculate $\text{var}_J, \text{cov}_{Jp}$ and a, b at the same stage. In the SSAT, we can calculate like the OP-SAT; however, the SSAT needs to compute vertical and horizontal filters; thus, this implementation of the guided image filtering using SSAT requires four stages.

The proposed implementation has high data locality; thus, the filtering is high cache efficiency. We can further improve the cache efficiency by tiling. The method of tiling divides an image into some tiles, and then processes tiles as one image. Since the size of the image become small, cache efficiency is improved. However we need padding for each tile, then we need redundant calculation.

IV. EXPERIMENTAL RESULTS

We compared five types of scheduling of the guided image filtering, which are SSAT without tiling, OP-SAT without tiling, SSAT with tiling, OP-SAT with tiling and conventional implementation. Filtering radius was $r = 2$, and image size were $512 \times 512, 512 \times 768$ and 1980×1080 . We used the 1ch image for input and guidance image. We computed by Intel Core i7 5960X (8 cores, 16 threads) with Visual Studio 2017. We used OpenMP and AVX/AVX2 optimization for the experimental codes.

TABLE I shows computational times for each schedule. The result when tiling is that uses the best result when dividing the number of tiles from 2 ($1 \times 2, 2 \times 1$) to 4096 (64×64), which is vertical \times horizontal. Conventional implementation is the OpenCV 3.0's implementation. As a result, without tiling, the OP-SAT is faster than the SSAT. Then with tiling, the SSAT is faster than the OP-SAT.

V. CONCLUSION

In this paper, we proposed efficient computational scheduling of the guided image filtering, which has high cache efficiency using fast box filtering and tiling. We divided input and guidance images into tiles, then use the guided image filtering with the SSAT or OP-SAT box filtering for each tile. Our methods using the SSAT with tiling is about 30 times faster than the conventional one.

REFERENCES

- [1] K. He, J. Shun, and X. Tang, "Guided image filtering," in *Proc. European Conference on Computer Vision (ECCV)*, 2010, pp. 1–14.
- [2] F. C. Crow, "Summed-area tables for texture mapping," in *Proc. ACM SIGGRAPH*, 1984, pp. 207–212.
- [3] N. Fukushima, Y. Maeda, Y. Kawasaki, M. Nakamura, T. Tsumura, K. Sugimoto, and S. ichiro Kamata, "Efficient computational scheduling of box and gaussian fir filtering for cpu microarchitecture," in *Proc. Asia-Pacific Signal and Information Processing Association*, 2018.