International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

A Filter-Driven Integral Image Generation Method for Scalable Image Resolution

Seongrim Choi[1] and Byeong-Gyu Nam*[2]

Submitted: 12/03/2024 **Revised:** 27/04/2024 **Accepted:** 04/05/2024

Abstract: Integral images are widely used for computer vision applications such as face detection and object recognition because they can be utilized to speed up the feature computation step. In recent years, there has also been an increasing demand for the use of integral images in high-resolution computer vision applications. However, integral images need a significant amount of memory space since it exploits a large word length to represent the accumulation for filtering operations. There have been studies on the size reduction of the integral images such as word length reduction and partial accumulation methods. However, these approaches were not suited for high-resolution applications because their memory usage increases rapidly following the image resolution. Therefore, in this letter, we present a filter-driven integral image generation method for scalable integral image resolution. The proposed method generates integral images following the filter height which has much smaller dimension than the image resolution that the previous studies used. Consequently, the proposed filter-driven method is less affected by the image resolution of target applications. Evaluation results show the proposed method is scalable up to ultra-high definition (UHD) by reducing the memory usage by 76.4% compared with the state-of-the-art.

Keywords: *integral image, computer vision, face recognition, object recognition, image resolution*

1. Introduction

Integral image was originally introduced as the summed-area table for texture mapping in computer graphics [1], and is now widely adopted in computer vision applications [2]. It is a data structure for quickly and efficiently generating the sum of values in a rectangular subset of a grid and thus, used for face detections and object recognition algorithms to speed up the feature computation step. Recently, there is an increasing demand for the use of integral images in high-resolution applications because features need to be detected in the overlapping or occlusion conditions in crowded environments [3]. However, the integral image needs a significant amount of memory space since its pixel values are generated by accumulating all the previous pixels in the input image.

There have been studies on reducing the size of integral images [4-8]. A word length reduction scheme exploits modulo operation and rounding to reduce the word length of integral images [4, 5], but it incurs rounding errors. This problem was solved by a partial accumulation method that does not use the rounding but accumulates only a part of pixel values rather than all the previous pixels in producing integral image pixels [6]. However, these methods were not suited for recent high-resolution applications because they rapidly increase the memory usage following the image resolution. Dynamic integral image generation methods store the integral image for a small size filter rather than that for the entire image [7, 8]. However, they require as many memory blocks as the vertical length of the filter to dynamically produce an integral image of the filter within a single clock cycle. Therefore, these approaches are not suitable for modern studies accommodating large filters with long vertical lengths [9].

In this letter, we present a filter-driven integral image generation method to support the scalable image resolution. The method stores the integral image following the filter height that has much smaller dimension than the image size which is used by the previous studies and thus, is less affected by the image resolution. As a result, the proposed method is scalable up to the ultra-high definition (UHD) by saving its memory space by 76.4% from the state-of-the-art [6].

2. A Filter-Driven Method for Producing Integral Images

Conventionally, integral images are produced according to the Eq. (1). Each pixel in the integral image is calculated by accumulating all the input pixels residing before the current pixel in the input image. Therefore, its memory size becomes very large to represent the accumulation of the input pixels. Moreover, the memory size grows rapidly regarding the input resolution because of the intensive pixel accumulation made in the high resolution integral images.

$$ii(x,y) = \sum_{x'=0}^{x} \sum_{y'=0}^{y} i(x',y') \quad 0 \le x < W, 0 \le y < H \qquad (1)$$

where i (x', y') is the pixel value at position (x', y') of an input image i, ii (x, y) is the pixel value at position (x, y) of an integral image ii, and W and H are the width and the height of the input image i, respectively.

Therefore, we introduce a scalable method in producing the integral images that is less affected by the image resolution. Conventionally, an integral image takes memory space of $W \times H$, where W and H are the width and height of input image,

[1] *Dept. of Computer Science and Engineering, Chungnam National University, 99, Daehak-ro, Yuseong-gu, Daejeon, 34134, Korea*
ORCID ID : 0000-0003-3144-3616
[2] *Dept. of Computer Science and Engineering, Chungnam National University, 99, Daehak-ro, Yuseong-gu, Daejeon, 34134, Korea*
ORCID ID : 0000-0003-0069-1959
* *Corresponding Author Email: bgnam@cnu.ac.kr*

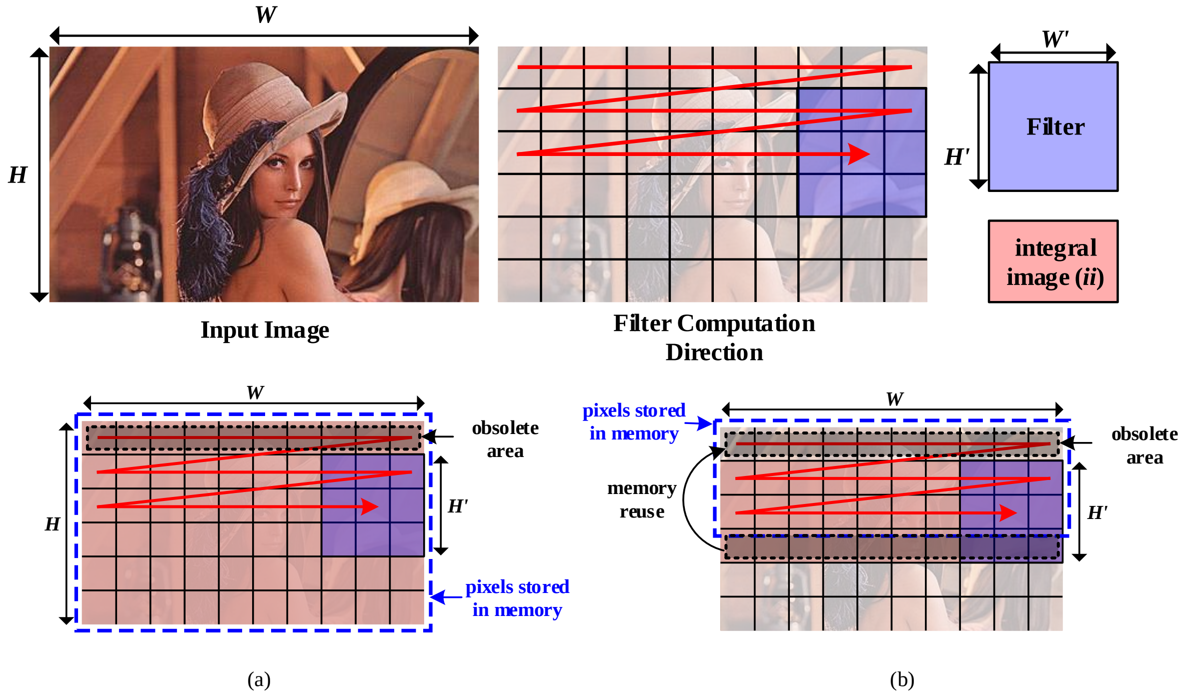

Fig 1. Integral image pixels stored in memory space (a) conventional scheme (b) proposed scheme

respectively, as illustrated in Figure. 1a. Note that the integral image pixels that finished filtering are not used anymore during the filtering process that proceeds with a sliding window manner from the top-left to the bottom-right of the input image. Therefore, the proposed method overwrites the newly produced pixels to the memory location containing the obsolete pixels, as described in Figure. 1b. As a result, the integral image from the proposed method takes smaller memory space of $W \times H$' compared with that from the state-of-the-art [6] which takes memory space of $W \times H$ because filter height H' is generally far smaller than H. Generally, row integral values of the pixels corresponding to H' are stored rather than the column integral values since typical systems receive an input image row by row as shown in Figure. 1a. However, the memory space can be reduced further to $W' \times H$ in case a system receives the input image in column-wise manner because reducing the width has higher impact than reducing the height in memory layout with typical aspect ratio of larger width than height.

Algorithm 1: The main procedure of the proposed integral image generation method

Input: i (x, y): the pixel value at position (x, y) of an input image i; W and H: the width and the height of an input image i, respectively; W' and H': the width and the height of a filter f;

Variables: ii (x, y): the pixel value at position (x, y) of an integral image ii; rii (x, y): the pixel value at position (x, y) of a row integral image rii;

memory address of $ii \leftarrow 0$;
$x \leftarrow 0, y \leftarrow 0$;
while $(y < H)$ **do**
while $(x < W)$ **do**
  $ii\ (x, y) \leftarrow ii\ (x, y-1) + rii\ (x-1, y) + i\ (x, y)$;
  if (memory address of ii == $(W \times H'$-1)) **then**
    memory address of $ii \leftarrow 0$;
  else
    memory address of ii++;
  x++;
  if $(x \geq (W'$-1) && $y \geq (H'$-1)) **then** $f(\cdot)$;
end while
$x \leftarrow 0, y$++;
end while

Algorithm 1 shows the main procedure of the proposed integral image generation method. An integral image is produced by incrementing the position variables x and y from the top-left to the bottom-right of the input image, and a filtering takes place when the integral image is prepared for the filtering i.e. when it gets larger than the filter dimension. Conventionally, an integral image pixel ii (x, y) is produced by adding the input pixel i (x, y), integral image pixel ii $(x, y$-1), and the row integral image pixel rii $(x$-1, $y)$ which is the cumulative row sum of all input pixels up to the left of the i (x, y) [2]. Meanwhile, the memory address of an integral image pixel is counted up until it reaches the address of $W \times H$-1. On the contrary, the proposed method initializes the memory address of the integral image at $W \times H$'-1 to zero, thereby overwriting the newly computed pixels to the memory location of the obsolete pixels. This approach saves memory usage by 83.23% for the UHD resolution compared with the conventional method [2].

3. Evaluation Results

As shown in Figure. 2, we investigate the memory usage of the integral image following both the filter size and image resolution. We use the full-high definition (FHD) and UHD images as test image resolutions because these are widely adopted in recent computer vision systems. Even though the memory usage of the proposed method increases following the filter size as shown in Figure. 2a, it is still far less than that of previous studies. In order to evaluate the memory usage regarding the image resolution, we adopt the 387×387 filter which is the largest filter dimension generally adopted in integral images [9]. As shown in Figure. 2b, the memory usage of the proposed scheme shows modest increase regarding the image resolution, whereas the memory usages increase rapidly for the previous approaches. Memory space is saved by 52.3% and 76.4% for the FHD and UHD, respectively, from this work compared with that from the state-of-the-art [6].

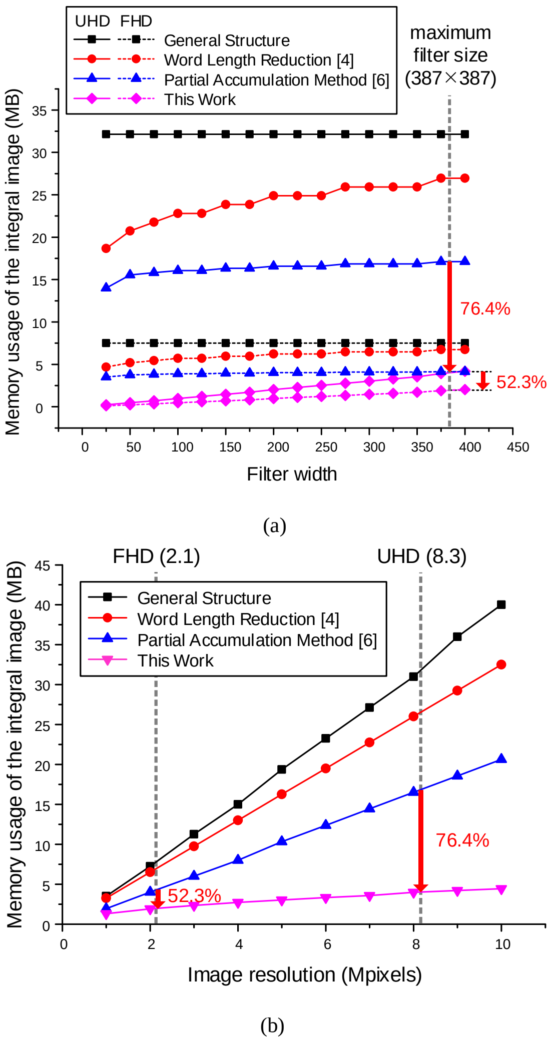

Fig 2. Memory usage comparison of the integral image following (a) the filter size and (b) the image resolution. Note: filter height equals to filter width.

4. Conclusion

We propose a filter-driven integral image generation method for scalable image resolution. In the proposed method, the memory usage is less affected by the image resolution because the integral image is produced following the filter height rather than the input image resolution. The proposed method is scalable up to the UHD by reducing memory space by 76.4% compared with the state-of-the-art.

Acknowledgements

This work was supported by research fund of Chungnam National University. The authors would like to thank the IC Design Education Center(IDEC) for EDA tool support.

References

[1] F. C. Crow, "Summed-area tables for texture mapping," *ACM SIGGRAPH Computer Graphics In Proceedings of SIGGRAPH*, 1984, 18, (3), pp. 207-212

[2] P. Viola and M. Jones, "Robust real-time object detection," *International Journal of Computer Vision (IJCV)*, July, 2001. pp. 1-25

[3] O. David et al., "Real-time GPU-based face detection in HD video sequences," *IEEE Int. Conf. Computer Vision (ICCV)*, Nov. 2011, pp. 530-537

[4] H. J. W. Belt, "Word length reduction for the integral image," *in Proc. IEEE Int. Conf. Image Processing (ICIP)*, Oct. 2008, pp. 805-808

[5] J. Kim et al., "Low-cost hardware architecture for integral image generation using word length reduction," *in Proc. International Symposium on Circuits and Systems (ISOCC)*, Oct. 2020, pp. 119-120

[6] S.-H. Lee and Y.-J. Jeong, "A new integral image structure for memory size reduction," *IEICE Trans. Inf. & Syst.*, 2014, E97-D, (4), pp. 998-1000

[7] C. Kumar and S. Agarwal, "A novel architecture for dynamic integral image generation for Haar-based face detection on FPGA," *TENCON 2014-2014 IEEE Region 10 Conference, Bangkok*, Oct. 2014, pp. 1-6

[8] D. Kim et al., "Memory-efficient architecture for contrast enhancement and integral image," *Int. Conf. Electron. Inf. Commun (ICEIC)*, Jan. 2020, pp. 1-4

[9] D. Jeon et al., "An energy efficient full-frame feature extraction accelerator with shift-latch FIFO in 28 nm CMOS," *IEEE J. Solid-State Circutis*, 2014, 49, (5), pp. 1271-1284