A Configurable Parallel Hardware Architecture for Efficient Integral Histogram Image Computing

Abstract:

Integral histogram image can accelerate the computing process of feature algorithm in computer vision, but exhibits high computation complexity and inefficient memory access. In this paper, we propose a configurable parallel architecture to improve the computing efficiency of integral histogram. Based on the configurable design in the architecture, multiple integral objects for integral histogram image, such as image intensity, image gradient, and local binary pattern, are well supported. Meanwhile, by means of the proposed strip-based memory partitioning mechanism, this architecture processes the integral histogram quickly with maximal parallelism in a pipeline manner. Besides, in this architecture, the proposed data correlation memory compression mechanism effectively solves the expansion problem of integral histogram memory caused by storing the histogram data. It fully reduces the data redundancy in the integral histograms, and saves a lot of memory resources. The proposed architecture of this paper analysis the logic size, area and power consumption using Xilinx 14.2.

Enhancement of the project:

Existing System:

In vision applications, the integral histogram image is a generic concept, which can be used for accelerating different feature algorithms by using different integral objects. If the integral object is the image intensity, it can be used to not only accelerate the Haar feature computing in the face detection applications, but also accelerate the scale-invariant feature transform (SIFT) feature and speedup robust feature (SURF) feature. If the integral object is the gradient of the image pixels, the histogram of gradients feature can be computed quickly. If the integral object is the local binary pattern (LBP), the LBP-based vision application can be accelerated.

However, with the advance of camera technology, the increasing image size poses a major challenge to vision applications. The computation complexity of integral histogram is getting high. As we have mentioned above, the integral histogram computing is characterized by high
data dependence and frequent memory access. In order to improve the computing speed of integral histogram in practical implementations, some previous works aim to increase performance by using multicore platform and graphic processing unit (GPU) platform. Implementations on multicore and GPU platform can support generic integral histograms mentioned above by programming, but the power efficiency of these schemes are not satisfactory. Some works design specific hardware architectures for the integral histogram to achieve an efficient computation. However, these works are specifically designed for a certain kind of integral object and the generic integral histogram cannot be supported. Meanwhile, the memory expansion problem is not considered. As the image size becomes larger, the computing efficiency gets lower.

Histogram image is very useful for many computer vision applications. In order to improve the histogram computing method, many fast computation methods are proposed. This method exploits the previous computed histograms and updates the new histograms by added new column of histograms and removed old ones. Meanwhile, some parallel hardware architecture are proposed to accelerate histogram computing. However, many computer vision applications require the summarization of histogram values in specific regions, and shows the characteristic of no overlap of sliding window for reusing the previous histograms. The integral histogram technique proposed in can compute the summarization of histograms in a specific region in constant time. Hence, it is more useful in such kinds of applications.

Disadvantages:

- computing efficiency is low
- high area and power

Proposed System:

Overall Architecture
As shown in Fig. 1, our proposed architecture consists of controller unit (CU), OIM, integral histogram calculation unit (CALU), integral compression unit (ICU), integral restore unit (IRU), and integral histogram memory (IHM). CU writes the input image data into OIM. Under the control of CU, the CALU loads the data from OIM to achieve the integral histogram computing.

Integral Histogram Computing Architecture Design

As shown in Fig. 2, the core computing unit CALU mainly consists of the configurable operation unit (COU) and a set of w-structures. w-structure denotes a size of $w \times \lceil \log_2(w - 1) + 1 \rceil$ computing array.
Fig. 2. Architecture of the CALU.

Memory Partitioning Mechanism

In order to support the maximal parallelism-based pipeline computing in CALU, OIM must support the fetching of w data simultaneously. Hence, we design a strip-based memory partitioning mechanism. As shown in Fig. 3(a), the m × n image is divided to several w × n image strips. While designing the hardware architecture, given a strip width w, we construct the w-structure in CALU and design pipeline processing mechanism to adapt to the input of different w × n image blocks, as shown in Fig. 3(b).
Fig. 3. Strip-based memory partitioning. (a) Image is divided into strips. (b) Strips are processed in pipeline way.

Advantages:

- improve the computing efficiency
- improve area and power

Software implementation:

- Modelsim
- Xilinx ISE