Microprocessors and Microsystems 38 (2014) 458–469

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

Accelerating image boundary detection by hardware parallelism

Zhilei Chai *, Xinglong Shao, Yuanpu Zhang, Wenmin Yang, Qin Wu

Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), Department of Computer Science, Jiangnan University, Wuxi, Jiangsu Province 214122, China

ARTICLE INFO

Article history:
Available online 18 April 2014

Keywords:
Image boundary detection
Hardware acceleration
Parallel computing
Real-time performance
Field programmable gate arrays (FPGAs)

ABSTRACT

Image boundary can provide useful information for high-level tasks in computer vision applications. However, high-quality image boundary detection algorithms are computationally intensive, which limits their applicability in real-world applications. In this paper, a study on accelerating algorithms of image boundary detection by hardware parallelism is presented. The Pb (Probability boundary) algorithm, as one representative high-quality algorithm of gradient-based boundary detection, is selected. Firstly, different types of parallelisms existing in Pb are analyzed. Then, suitable hardware structures to accelerate Pb based on those parallelisms are discussed. Finally, time performance, accuracy and scalability of the parallel Pb detector accelerated by hardware are presented. After being implemented in a Xilinx Virtex-7 FPGA, XC7VX485T-2FFG1761C, the parallel Pb detector with the working frequency of 200 MHz takes 6.3 ms to process an 321 × 481 image. It is more competitive than Pb implemented on CPUs when larger images are processed. This paper demonstrates a promising way to improve the real-time performance of high-quality image boundary detection systems, especially when embedded and real-time systems are taken into account.

1. Introduction

Techniques of image boundary detection are widely used in computer vision applications including image segmentation, object recognition and classification etc. In contrast to an edge that is often defined as an abrupt change in low-level image features such as brightness or color, a boundary is a contour that represents a change in pixel ownership from one object or surface to another [1]. Boundary detection is thus a high-level technique based on edge detection that provides more instructive information but with higher computational cost. For instance, as a representative gradient-based algorithm for high-quality boundary detection, the Pb (Probability boundary) detector takes more than 1 min to process a 321 × 481 image when being executed on a 2.10 GHz Pentium (R) dual-Core CPU T4300. That is why most real world applications today still use simpler, less accurate boundary detection techniques rather than high quality techniques with high computational complexity. For vision systems used in embedded applications, the real-time performance is more critical because those systems usually need to interact with real world. Thus, to accelerate computation of boundary detection is vital for deploying high-quality methods in real world applications.

Due to the fact that frequency scaling of processors becomes more difficult, researchers today pay more attention to accelerate computation by exploiting parallelisms fully through architectural optimization. For instance, Bryan Catanzaro et al. [3] investigated how to accelerate boundary detection by implementing it on parallel hardware of GPUs. The advantage of implementing boundary detection algorithms on GPUs is high programmability. The disadvantage is that GPUs have to be cooperated with CPU and the system based on GPUs has relatively higher power consumption. Thus, more efficient approaches to accelerate boundary detection should be investigated.

In this paper, a study on accelerating algorithm of Pb boundary detection on an FPGA (Field Programmable Gate Arrays) platform [4–6] is presented in order to exploit different types of parallelisms fully. After being implemented in a Xilinx Virtex-7 FPGA, XC7VX485T-2FFG1761C, the parallel Pb detector with the working frequency of 200 MHz takes 6.3 ms to process an image of 321 × 481, in contrast to 0.67 s on GPUs and 72.5 s on CPUs.

The main contribution of this work is summarized as follows:

- A comprehensive analysis of different types of parallelisms existing in Pb including task parallelism, data parallelism and pipeline parallelism is conducted. It provides a general guidance for users to implement parallel boundary detection systems.

* Corresponding author.
E-mail addresses: zlchai@jiangnan.edu.cn (Z. Chai), qinwu@jiangnan.edu.cn (Q. Wu).

http://dx.doi.org/10.1016/j.micpro.2014.03.011
0141-9331/

- A representative gradient-based algorithm for high-quality boundary detection is implemented in FPGA, and its scalability, usability are discussed. Because this algorithm includes many fundamental modules used in gradient-based boundary detection methods, this study could provide a foundation for other gradient-based boundary detection algorithms to be accelerated by hardware.
- Through presenting the acceleration effect by implementing Pb in FPGA, this paper demonstrates a promising way to improve the real-time performance of high-quality image boundary detection systems.

The rest of the paper is organized as follows. Section 2 introduces related work on boundary detection algorithms and their acceleration. Section 3 gives an general overview of gradient-based boundary detection methods. Section 4 introduces processes of the Pb algorithm in more detail. Section 5 analyzes parallelisms existing in Pb at different types and levels. Section 6 introduces the hardware design and corresponding experiments for the parallel Pb detector. Section 7 shows overall experimental results and discusses time performance, accuracy and scalability of the hardware-accelerated Pb detection system. Section 8 concludes the paper.

2. Related work

Although there are different edge/boundary detection techniques [1,7,8,15,16], recent development of the gradient-based technique [1,15] makes it more compelling. In this section, related work on algorithm investigation of gradient-based boundary detection is introduced first. Then, efforts related to computing acceleration for gradient-based boundary detection are introduced.

2.1. Gradient-based boundary detection algorithms

A variety of edge detectors were developed based on image gradient or Gaussian derivative filters, such as the popular Canny edge detector [9] that is still used today. For gradient-based methods, the brightness-only model cannot detect texture boundaries well. Thus, it is a trend to combine changes of multiple features, such as brightness, color, and texture, together to improve boundary detection effect.

Martin et al. [1] presented an algorithm that uses multiple cues of brightness, color, and texture respectively to detect natural image boundaries. It checks each pixel for local discontinuities in different feature channels. For instance, brightness gradient (BG), color gradient (CG), and texture gradient (TG) are taken into account simultaneously. For each feature channel, gradient information of multiple orientations is considered further for the discontinuity. After combining different cues together and using some optimization technique, the pixel with certain probability to be the boundary is selected as a boundary pixel. Thus, the algorithm is called Pb (i.e., Probability boundary) algorithm.

Dollar et al. [10] proposed a supervised learning algorithm for edge and object boundary detection, referred to as boosted edge learning (BEL). The decision of an edge point is made independently at each location of the image. A very large aperture is used to provide significant context for decision. During the learning stage, the algorithm selects and combines a large number of features across different scales, in order to learn a discriminative model using an extended version of the probabilistic boosting tree classification algorithm. The learning-based framework is highly adaptive and there is no parameter need to tune.

Through an empirical study, Ren et al. [11] demonstrated multi-scale schemes to improve boundary detection results on large datasets of natural images. It utilized local boundary cues including contrast, localization and relative contrast, and trained a classifier to integrate them across multiple scales. The approach successfully combined strength from both large-scale detection (robust but poor localization) and small-scale detection (detail-preserving but sensitive to clutter) together.

M. Maire et al. [15] developed a new high-quality boundary detector using a combination of both local and global cues, called gPb (i.e., global probability boundary). It detects and localizes candidate junctions, takes into account both boundary salience and geometric configuration. As measured by the Berkeley segmentation dataset [24], the gPb detector is the highest quality detector for natural image boundary detection to date. Notably, gPb is based on Pb but its global processing is not so well suited to be parallelised as that of local. So in this paper, Pb is selected for acceleration to provide a common foundation for hardware acceleration of gradient-based boundary detection algorithms.

2.2. Computing acceleration of image boundary detection

As mentioned above, the boundary detection algorithm with better detection results means more information being taken into account and higher computational cost. Thus, the real-time performance is increasingly becoming a critical factor for high quality boundary detection techniques when being used in real applications. In order to deal with this issue, Ray Hidayat et al. [2] introduced a real-time texture boundary detector, running on an Intel Core 2 Duo 2.66 GHz CPU. When image resolution is 320×240, 43.29 fps can be processed in time. Please note that only texture is considered in their experiment, brightness and color are not included. Due to the sequential model of computation and the constraint of frequency increasing, CPUs are difficult to accelerate advanced boundary detection algorithms further.

Catanzaro et al. [3] optimized and implemented gPb/Pb on Nvidia GPUs for higher processing speed. The computational time of the boundary detection was decreased from 236.4 s per frame on general purpose CPUs to 1.822 s on Nvidia GPUs for images with the size of 321×481. Furthermore, the original gradient computation of gPb/Pb was replaced by a simpler method in their implementation.

To our knowledge, there is few work focusing on optimizing high-quality boundary detection algorithms on FPGAs. Only implementations of simple edge detectors or fundamental modules used in boundary detection on FPGAs were introduced [12–14,20,26–28]. In this paper, Pb is selected as a representative to be optimized on FPGAs in order to study key techniques for accelerating high-quality boundary detection algorithms based on hardware parallelism.

3. An overview of gradient-based image boundary detection algorithms

As shown in Fig. 1, although lots of different gradient-based boundary detection algorithms exist, they are generally composed of the following steps as options.

- Color space conversion. To convert color space from RGB to Lab, Gray-scale or others.
- Gradient computation. To compute necessary gradient information from different image features such as brightness, color (sometimes multiple channels), texture or others.
- Cue combination. Use techniques such as logistic regression to combine cues computed from different features together on multiple orientations or scales.
- Global cue computation. Use techniques such as normalized cuts to obtain global information of cues based on combined local cues.