J. Parallel Distrib. Comput. 92 (2016) 1–14

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

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

Parallelizing image feature extraction algorithms on multi-core platforms

Yunping Lu [d,b], Yi Li [a,b,c], Bo Song [a,b,c], Weihua Zhang [a,b,c,*], Haibo Chen [e], Lu Peng [f]

[a] Software School, Fudan University, Shanghai, China
[b] Shanghai Key Laboratory of Data Science, Fudan University, Shanghai, China
[c] Parallel Processing Institute, Fudan University, Shanghai, China
[d] School of Computer Science, Fudan University, Shanghai, China
[e] Institute of Parallel and Distributed Systems, Shanghai Jiao Tong University, Shanghai, China
[f] Division of Electrical and Computer Engineering, Louisiana State University, United States

HIGHLIGHTS

- Analysis and evaluation of various parallelism in image feature extraction algorithms.
- Observations on parallelism constraints in image feature extraction algorithms.
- An efficient adaptive pipeline scheme with good scalability.
- A power-efficient parallelism algorithm for various workloads.

ARTICLE INFO

Article history:
Received 11 October 2014
Received in revised form
10 August 2015
Accepted 2 March 2016
Available online 10 March 2016

Keywords:
Image feature extraction
SIFT
SURF
Adaptive pipeline
Multi-core

ABSTRACT

Currently, multimedia data has become one of the most important data types processed and transferred over the Internet. To extract useful information from a huge amount of such data, SIFT and SURF, as two most popular image feature extraction algorithms, have been widely used in many applications running on multi-core platforms. However, limited parallelism in existing designs makes it hard or impossible to apply them in many applications with real-time requirements. Therefore, it has become one of the major challenges to improve the processing speed of image feature extraction algorithms.

In this paper, we first analyze the parallelism constraints in the algorithms, such as imbalanced workloads and indeterminate time distributions. Based on such analyses, we present an adaptive pipeline parallel scheme (AD-PIPE) to adjust the thread number in different stages according to their workloads dynamically, which achieves a balanced partition for constant input workloads. Furthermore, we also implement a power efficient version (AE-PIPE) for AD-PIPE through scheduling threads based on variable input workloads. Experimental results show that AD-PIPE achieves a speedup of 16.88X and 20.33X respectively over SIFT and SURF on a 16-core machine. Moreover, AE-PIPE achieves up to 52.94% and 58.82% power saving with only 3% performance loss.

1. Introduction

As our society has entered a data-centric world, a huge amount of data is transferred and processed over the Internet. As indicated in the forecast of CISCO Inc. [8], until 2014, the data quantity generated every month will reach more than 0.6 million PB. Among such a huge amount of data, multimedia data has become one of the most common data types being processed.

With the dramatic increase of multimedia data, it is vitally important to continually collect, index and retrieve such ever-increasing data to extract useful information and understand them. Therefore, image feature extraction algorithms (IFEAs), as the fundamental components of image and video retrieval applications, have been designed and developed for many years. Among the IFEAs, SIFT (*Scale Invariant Feature Transform*) [20,21]

* Corresponding author at: Software School, Fudan University, Shanghai, China.
E-mail addresses: luyping@sina.com (Y. Lu), yee.lie@gmail.com (Y. Li), espiesong@gmail.com (B. Song), zhangweihua@fudan.edu.cn, whzhang.fd@gmail.com (W. Zhang), haibochen@sjtu.edu.cn (H. Chen), lpeng@lsu.edu (L. Peng).

http://dx.doi.org/10.1016/j.jpdc.2016.03.001
0743-7315/

and SURF (*Speeded Up Robust Features*) [4] are two most popular ones [25,2]. They have been widely used in many applications, such as image and video retrieval [18,27], object recognition [15,3], and face recognition and authentication [7,10].

However, the processing speed of the current image feature extraction algorithms (IFEAs) still has a large room to be improved. For example, SIFT can only process about 1.8 images or frames per second on an average CPU and SURF can only process about 2.6 images or frames per second based on our experiments. Such results make them impossible to be used in scenarios with real-time requirements, such as large-scale content-based image retrieval or object recognition systems. The major reason stems from their design complexity: to make the algorithms insensitive to rotation, scaling, contrast and viewpoint changing, some complex transformations are included in the design of the IFEAs. Moreover, to guarantee retrieval accuracy, hundreds or thousands of feature points are extracted to represent an image or a frame. Each feature point will be further described with information around it and filled into a high-dimensional vector. Therefore, the algorithms are not only computation intensive but also data intensive.

The popularity of multi-core architecture and the increase of computation resources on such platforms provide a new opportunity to accelerate the processing speed of the IFEAs. Such opportunity has been evidenced by recent efforts on parallelizing these algorithms, such as [14,35,36]. However, the achieved speedup can still be further improved. For example, the parallelizing effort in [37,29] only achieves a speedup of about 6X on a 16-core machine.

In this paper, we first systematically analyze the characteristics related to parallelization of the IFEAs (SIFT and SURF). We find that there exist some inherent parallel constraints, such as imbalanced workloads in different feature points, sub-blocks, images, and indeterminate time distributions of different functions. To alleviate these constraints, we design and implement an adaptive pipeline parallelism scheme (AD-PIPE) for the IFEAs with constant input workloads. AD-PIPE partitions different functions into different pipeline stages in a producer–consumer manner. Such a design can efficiently overcome the constraints of imbalanced workloads. To alleviate the constraint of indeterminate time distributions among stages, we apply an adaptive strategy. The strategy can dynamically check workloads in different pipeline stages and adjust the thread number in different stages to achieve a balanced partition. To further improve AD-PIPE from the perspective of both performance and energy, we also extend AD-PIPE to design and implement a power-efficient version for variable input workloads, called AE-PIPE. AE-PIPE can adjust the number of threads between the working state and the idle state based on input workloads to reduce unnecessary computation resources.

Experimental results show that such designs are efficient and scalable. AD-PIPE can achieve a speedup of 16.88X and 20.33X respectively over SIFT and SURF on a 16-core commodity machine and a real-time processing speed of about 30 and 52 images or frames per second. For AE-PIPE, it can achieve about 23.81% and 25.30% power saving respectively over SIFT and SURF for variable input workloads, and up to 52.94% and 58.82% power saving for constant workloads with only 3% performance loss.

In summary, the main contributions of this paper can be summarized as follows.

- An analysis and an evaluation of various parallelism in the IFEAs, including image level, block level, scale level, pipeline level and their combinations.
- The observations on parallelism constraints in the IFEAs, including the imbalanced workloads and indeterminate time distributions.
- The design and implementation of an efficient adaptive pipeline scheme for the IFEAs with constant input workloads, which is suitable for the IFEAs and outperforms prior designs with good scalability.
- The design and implementation of a power-efficient parallelism for the IFEAs with variable input workloads, which can schedule threads based on input workloads.

The paper is organized as follows. Section 2 gives an overview of various IFEAs and related acceleration work. Section 3 describes SIFT and SURF in brief and presents a systematic characteristics analysis of them. Section 4 focuses on the design and implementation of our adaptive pipeline parallel scheme. Adaptive power-efficient pipeline parallelism is discussed in Section 5. In Section 6, we show detailed evaluation results on multi-core platforms. Finally, we conclude our work in Section 7.

2. Related work

In this section, we first introduce some image feature extraction algorithms (IFEAs). Then, we discuss some previous acceleration efforts on the IFEAs.

2.1. Image retrieval algorithms

As the fundamental components of image and video retrieval applications, image retrieval algorithms extract features to represent an image (or a video frame). Image features usually can be divided into two categories: global feature-based and local feature-based.

Global feature-based algorithms (GFAs): GFAs usually use a single feature to represent an image or a video frame, such as color histogram and texture. Due to using only one feature to represent the entire image, GFAs fast extract the features but with low precision in matching (more than 30% error rate [28]), which limits their applications. Furthermore, the algorithms cannot be used to retrieval images after some transformations, such as re-sizing and cropping [5]. Thus in many applications, GFAs are insufficient.

Local feature-based algorithms: In contrast to GFAs, local feature-based algorithms (LFAs) extract hundreds or thousands of features to represent an image. To guarantee the retrieval accuracy, the algorithms generally include some complex computation steps. Due to high precision in matching, they have been widely used in many real-world applications.

Among all LFAs, SIFT and SURF are two most popular and robust ones [25,2]. Thus, we will mainly focus on them in this paper. SIFT [20,21] is the most publicly accepted and robust local feature-based image extraction algorithm. To meet different requirements, there are many variants of SIFT, such as GLOH [25], and PCA-SIFT [17]. Since all the variants are based on SIFT, in this paper, we only focus on the original SIFT algorithm for research. Another widely-used algorithm is SURF [4]. After proposed in 2006, it has been applied to many applications, and tends to be an efficient alternative of SIFT. Both of two algorithms are insensitive to various transformations, such as scaling, rotation and illumination.

2.2. Previous acceleration efforts

Parallelization, as an efficient approach, has been widely used in different areas for performance acceleration, such as atmosphere prediction [32,31,19], graph processing [34], and architecture simulation [30]. Since the LFAs include complex computations and have to describe hundreds of feature points, they are time-consuming, which limits their application fields in the real world with real-time requirements. In order to solve the problem, many efforts have been done to accelerate SIFT and SURF through exploiting inherent parallelism in them.