Procedia Computer Science

Volume 80, 2016, Pages 41–51

ICCS 2016. The International Conference on Computational Science

Acceleration of Tear Film Map Definition on Multicore Systems

Jorge González-Domínguez[1], Beatriz Remeseiro[2], and María J. Martín[1]

[1] Grupo de Arquitectura de Computadores, Universidade da Coruña
Campus de Elviña s/n, 15071 A Coruña, Spain
jgonzalezd@udc.es, mariam@udc.es

[2] INESC TEC - INESC Technology and Science
Campus da FEUP, Rua Dr. Roberto Frias, 4200–465 Porto, Portugal
bremeseiro@fe.up.pt

Abstract

Dry eye syndrome is a public health problem, and one of the most common conditions seen by eye care specialists. Among the clinical tests for its diagnosis, the evaluation of the interference patterns observed in the tear film lipid layer is often employed. In this sense, tear film maps illustrate the spatial distribution of the patterns over the whole tear film and provide useful information to practitioners. However, the creation of a single map usually takes tens of minutes. Medical experts currently demand applications with lower response time in order to provide a faster diagnosis for their patients. In this work, we explore different parallel approaches to accelerate the definition of the tear film map by exploiting the power of today's ubiquitous multicore systems. They can be executed on any multicore system without special software or hardware requirements. The experimental evaluation determines the best approach (on-demand with dynamic seed distribution) and proves that it can significantly decrease the runtime. For instance, the average runtime of our experiments with 50 real-world images on a system with AMD Opteron processors is reduced from more than 20 minutes to one minute and 12 seconds.

Keywords: Parallel Programming, Multithreading, Image Segmentation, Dry Eye

1 Introduction

Dry eye syndrome is a prevalent disease characterized by symptoms of ocular discomfort, ocular surface damage, reduced tear film stability, tear hyperosmolarity, and inflammatory components[3]. These features can be identified by different types of diagnostic tests. Among them, the grading of the tear film lipid layer appearance is usually the first clinical observation made by the experts [4]. This clinical test consists of two steps [5]: 1) acquiring an input image

Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2016

doi:10.1016/j.procs.2016.05.296

of the tear film lipid layer using the Tearscope Plus, an instrument which allows clinicians to rapidly assess the lipid layer thickness in a non-invasive way; and 2) categorizing this image into one of the five interference patterns defined for this purpose. This classification is a difficult clinical task, specially with thin layers which lack of color and/or morphological features. Consequently, there is a high degree of inter- and intra-observer variability.

In order to facilitate the task to the medical experts, by providing objective results and saving time, an automatic system for tear film classification was proposed in [10]. This research demonstrated that the interference patterns can be characterized by a feature vector composed of color and texture properties, and classified by means of machine learning algorithms. With the aim of reducing the computational requirements and allowing the classification system to work in real-time, feature selection was applied in [8]. That is, the feature vector can be now calculated in less than 1 second, while maintaining the accuracy over 96%.

Besides this global analysis, tear film maps were presented in [9] to represent the spatial heterogeneity of the tear film lipid layer and detect multiple interference patterns per image. This approach uses color and texture features as the global classification, and an adapted version of the seeded region growing algorithm to perform the segmentation of the image into the interference patterns. The proposed methodology is able to generate tear film maps with an accuracy over 90% in comparison with the manual annotations made by four experienced practitioners. However, there is still large room for improvement on processing time since the creation of tear film maps takes tens of minutes on a single CPU. Although the time needed to compute each feature vector is less than 1 second, the great number of feature vectors per single image leads the region growing step to a large processing time. This is nowadays a barrier for a wide adoption of this representation among experts, who usually expect shorter response time.

In this paper we present multithreaded approaches to accelerate the region growing step included in this algorithm by exploiting the computational power of multicore systems. Although parallel region growing implementations have been previously developed for multicore [6, 7, 11] and manycore systems [12, 14], the novelty of our work is three-fold: 1) we implement, for the first time, a parallel algorithm for the generation of tear film maps, which has different characteristics than the biomedical works presented on the state of the art; 2) we present a parallel version with dynamic seed distribution among threads that outperforms the static distributions applied in previous works, as will be shown in the experimental evaluation; and 3) up to our knowledge, this is the first work that provides an experimental evaluation of multithreaded region growing on multicore platforms with up to 64 cores.

The rest of the paper is organized as follows. Section 2 provides necessary background information. Section 3 describes the different parallel versions developed for tear film maps. Experimental results are presented in Section 4. Finally, Section 5 concludes the paper.

2 Background

This section presents the background of the problem, including a brief description of the classic algorithm for image segmentation known as seeded region growing. Additionally, its use to create tear film maps is also explained.

2.1 Seeded region growing

Given a set of initial points known as seeds, which can be manually or automatically selected, the algorithm finds a tessellation of the image into regions. The seeds are the first points of the regions, which are grown examining, through an iterative process, the neighboring points.