2015

Color and Edge Directivity Descriptor on GPGPU

Iakovidou, Chrysanthi

IEEE (Computer Society)

Color and Edge Directivity Descriptor on GPGPU

C. Iakovidou, L. Bampis, S. A. Chatzichristofis, Y. S. Boutalis and A. Amanatiadis,
Department of Electrical and Computer Engineering, Democritus University of Thrace, Greece
(ciakovid, loukbabi, schatzic, ybout, aamanat)@ee.duth.gr

***Abstract*— Image indexing refers to describing the visual multimedia content of a medium, using high level textual information or/and low level descriptors. In most cases, images and videos are associated with noisy and incomplete user-supplied textual annotations, possibly due to omission or the excessive cost associated with the metadata creation. In such cases, Content Based Image Retrieval (CBIR) approaches are adopted and low level image features are employed for indexing and retrieval. We employ the Colour and Edge Directivity Descriptor (CEDD), which incorporates both colour and texture information in a compact representation and reassess it for parallel execution, utilizing the multicore power provided by General Purpose Graphic Processing Units (GPGPUs). Experiments conducted on four different combinations of GPU-CPU technologies revealed an impressive gained acceleration when using a GPU, which was up to 22 times faster compared to the respective CPU implementation, while real-time indexing was achieved for all tested GPU models.**

I. Introduction

The area of Content Based Image Retrieval (CBIR) has seen a steady train of improvements in performance over the last decade [1]. The main focus of any CBIR method is to capture the rich information that images hold and vectorize it building its descriptor, so as to allow fast indexing and meaningful retrieval for the user.

The wide spread of affordable image and video capturing devices led to a rapid growth of multimedia databases in areas such as private life, journalism, medicine and tourist attraction, to name a few. Thus, in real life scenarios, description methods are not evaluated solely by their achieved performance but by their efficiency, as well.

Accelerating the descriptor-extraction procedure to the point where low level features can be described and incorporated in a file's header as it is captured and becomes part of a collection, is a matter of great importance. For instance, large image repositories such as *Flickr*, *facebook* and *Dropbox*, where millions of images are uploaded daily, will be able to index the images as they become part of their databases. The contribution of an implementation that achieves fast extraction of a descriptor is more evident when dealing with videos. According to 2014 YouTube's statistics[1], 100 hours of video are uploaded every minute. Automatic video annotation and summarization could be achieved through a descriptor extraction implementation that would allow real-time indexing of the frames.

Acknowledging the restrictions on computational resources that apply due to the massive amount of data involved in CBIR, recent methods are focused on producing compact vector representations. However, despite all the algorithmic efforts towards this direction, it is clear that useful acceleration without performance degradation can only be achieved through parallel processing.

Graphics Processing Units (GPUs) were first introduced to handle graphics primitives. However, the rapid evolution of the NVIDIA Compute Unified Device Architecture (CUDA) API [2], gave access to researchers from many varying fields, to the powerful parallel architecture of GPUs. Since GPUs are primarily employed for graphics, their configuration is ideal for parallelizing image processing algorithms [3].

Thus, multiple General-Purpose GPU (GP-GPU) implementations of existing image processing, indexing and categorization methods, have been proposed in recent literature. All report important acceleration and improved efficiency achieved by passing computations to the GPU, carefully following the architectural principles dictated by the GPU model.

In [4] the authors propose a compact histogram representation which applies replication and padding for optimizing the voting process in shared memory. They manage to minimize position conflicts by forcing adjacent threads to vote for different sub-histograms and propose the use of padding for reducing the amount of bank conflicts. A solution for building mosaic images of printed documents from frames selected from VGA resolution video captured from a mobile device, is presented in [5]. They utilize the device's GPU to perform the most demanding computations, concluding that the deep understanding of the data and the possible parallelizations becomes the crucial step for successful designs. A GPU implementation of Local Binary Pattern feature extraction is attempted in [6]. Results show that parallelizing a method that by design consists of processes that can be handled independently, ensures great achieved efficiency even when employing older graphics cards. However, in [7] authors attempt the same implementation on a mobile device's GPU and conclude that its parallelization power is insufficient. Taking a more general outlook, authors in [8] explored the design and implementation issues of image processing algorithms on GPUs. Selecting four major domains 3D shape reconstruction, feature extraction, image compression, and computational photography, they try to employ metrics that will allow the prediction of the effectiveness of a given image processing problem for the parallel implementation, and observed that speed-up varies extensively depending on the characteristics of each algorithm.

[1]https://www.youtube.com/yt/press/statistics.html

1066-6192/15 $31.00 © 2015 IEEE
DOI 10.1109/PDP.2015.105