CNN Based High Performance Computing for Real Time Image Processing on GPU

Sasanka Potluri, Alireza Fasih, Laxminand Kishore Vutukuru, Fadi Al Machot, and Kyandoghere Kyamakya
Transportation Informatics Group
Alpen-Adria University of Klagenfurt
Klagenfurt, Austria
{spotluri,lvutukur}@edu.uni-klu.ac.at
{Alireza.Fasih,Fadi.Almachot,Kyandoghere.Kyamakya}@uni-klu.ac.at

Abstract — **Many of the basic image processing tasks suffer from processing overhead to operate over the whole image. In real time applications the processing time is considered as a big obstacle for its implementations. A High Performance Computing (HPC) platform is necessary in order to solve this problem. The usage of hardware accelerator make the processing time low. In recent developments, the Graphics Processing Unit (GPU) is being used in many applications. Along with the hardware accelerator a proper choice of the computing algorithm makes it an added advantage for fast processing of images. The Cellular Neural Network (CNN) is a large-scale nonlinear analog circuit able to process signals in real time [1]. In this paper, we develop a new design in evaluation of image processing algorithms on the massively parallel GPUs with CNN implementation using Open Computing Language (OpenCL) programming model. This implementation uses the Discrete Time CNN (DT-CNN) model which is derived from originally proposed CNN model. The inherent massive parallelism of CNN along with GPUs makes it an advantage for high performance computing platform [2]. The advantage of OpenCL makes the design to be portable on all the available graphics processing devices and multi core processors. Performance evaluation is done in terms of execution time with both device (i.e. GPU) and host (i.e. CPU).**

Keywords— Image processing, Hardware accelerators, Cellular Neural Networks, GPUs, High Performance Computing, OpenCL

I. INTRODUCTION

Image processing is an ever expanding and dynamic area with applications reaching out into everyday life such as in medicine, space exploration, surveillance, authentication, automated industry inspection and in many more areas [3]. Real time image processing using modern processors is limited [4]. Problems in computer vision are computationally intensive [5]. The tremendous amount of data required for image processing and computer vision applications present a significant problem for conventional microprocessors [4]. Consider a sequence of images at medium resolution (512 x 512 pixels) and standard frame rate (30 frames per second) in color (3 bytes per pixel). This represents a rate of almost 24 million bytes of data per second. A simple feature extraction algorithm may require thousands of basic operations per pixels, and a typical vision system requires significantly more complex computations.

As we can see, parallel computing is essential to solve such problems [5]. In fact, the need to speed up image processing computations brought parallel processing into computer vision domain. Most image processing algorithms are inherently parallel because they involve similar computations for all pixels in an image except in some special cases [5]. Conventional general-purpose machines cannot manage the distinctive I/O requirements of most image processing tasks; neither do they take advantage of the opportunity for parallel computation present in many vision related applications [6]. Many research efforts have shifted to Commercial-Off-The-Shelf (COTS) -based platforms in recent years, such as Symmetric Multiprocessors (SMP) or clusters of PCs. However, these approaches do not often deliver the highest level of performance due to many inherent disadvantages of the underlying sequential platforms and "the divergence problem". The recent advent of multi-million gate on the Field Programmable Gate Array (FPGAs) having richer embedded feature sets, such as plenty on –chip memory, DSP blocks and embedded hardware microprocessor IP cores, facilitates high performance, low power consumption and high density [7].

But, the development of dedicated processor is usually expensive and their limited availability restricts their widespread use and its complexity of design and implementation also makes the FPGA not preferable. However, in the last few years, the graphic cards with impressive performance are being introduced into the market for lower cost and flexibility of design makes it a better choice. Even though they have been initially released for the purpose of gaming, they also find the scientific applications where there is a great requirement of parallel processing. Along with the support of hardware platforms there are some software platforms available like CUDA (Compute Unified Device Architecture) and OpenCL for designing and developing parallel programs on GPU [8]. Out of these available software platforms OpenCL framework recently developed for writing programs can be executed across multicore heterogeneous platforms. For instance, it can be executed on multicore CPU's and GPU's and their combination. Usage of this framework also provides an advantage of the portability that is; the developed kernel is compatible with other devices. Along with the available hardware and software platforms we used