Neurocomputing 151 (2015) 91–100

Contents lists available at ScienceDirect

Neurocomputing

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

A scalable CNN architecture and its application to short exposure stellar images processing on a HPRC

J. Javier Martínez-Alvarez *, Javier Garrigós **, Javier Toledo **, Carlos Colodro-Conde **, Isidro Villó-Pérez **, J. Manuel Ferrández **

Dpto. Electrónica, Tecnología de Computadoras y Proyectos, Universidad Politécnica de Cartagena, Cartagena, Spain

ARTICLE INFO

Article history:
Received 12 March 2014
Received in revised form
2 August 2014
Accepted 18 September 2014
Available online 31 October 2014

Keywords:
CNN
HLS
HDL
HPRC
HW/SW Co-execution
FPGA

ABSTRACT

A CNN-based algorithm for short exposure image processing and an application-specific computing architecture developed to accelerate its execution are presented. Algorithm is based on a flexible and scalable Cellular Neural Networks (CNN) architecture specifically designed to optimize the projection of CNN kernels on a programmable circuit. The objective of the proposed algorithm is to minimize the adverse effect that atmospheric disturbance has on the images obtained by terrestrial telescopes. Algorithm main features are that it can be adapted to the detection of several astronomical objects and it supports multi-stellar images. The implementation platform made use of a High Performance Reconfigurable Computer (HPRC) combining general purpose standard microprocessors with custom hardware accelerators based on FPGAs, to speed up execution time. The hardware/software partitioning and co-design process have been carried out using high level design tools, instead of traditional Hardware Description Languages (HDLs). Results are presented in terms of circuit area/speed, processing performance and output quality.

1. Introduction

Astronomy has achieved a great advance as a result of digital photography entering the stage. Hundreds of thousands of raw images can be processed and recombined to get a single sharper picture of an object of interest. Observation of astronomical transient events (short-lived phenomena) has also gained now a new impulse. As a result, the role of the astronomer has changed from being a sky observer to being a person dedicated to manage and process vast amounts of data. In response, an effort is being made to develop techniques and algorithms appropriate to the nature of problem and adapted to the characteristics of new computational tools, which, increasingly, can speed up calculations demanded by these instruments.

In this paper, a CNN-based algorithm for short exposure image processing (Lucky-Imaging techniques) and an application specific computing architecture developed to accelerate the algorithm are presented. The computing platform is a High Performance Reconfigurable Computer (HPRC), also called hybrid supercomputer, combining general purpose standard microprocessors with custom hardware accelerators based on Field Programmable Gate Array (FPGA) [1]. Hardware/software co-execution and partitioning of the algorithm bring the best of both, general purpose microprocessors flexibility and dedicated CNN kernels parallelism, to speed up execution time. The proposed algorithm is adequate for image registration methods used in short exposure observational techniques and makes intensive use of CNN kernels at the pre-processing and post-processing stages. In this context, the use of CNNs [2] provides the advantage of their versatility, which allows the modification of the algorithm and, through suitable training, the adjustment of the instrument to each application, i.e., the detection of different astronomical objects or events. The proposed algorithm improves the spatial resolution of multi-stellar images reaching the diffraction limit of the telescope.

The hardware specification of the CNN has been carried out using High-Level Synthesis (HLS) languages, instead of traditional Hardware Description Languages (HDLs). This design methodology provides hardware designs from high level descriptions, using several flavors of well-known programming languages, such as C, C++ or Matlab. Both the HLS tool and the HPRC platform were successfully evaluated previously in [3,4], where they were used to

* Principal corresponding author.
** Corresponding authors.
E-mail addresses: jjavier.martinez@upct.es (J.J. Martínez-Alvarez), javier.garrigos@upct.es (J. Garrigós), javier.toledo@upct.es (J. Toledo), carlos.colodro@upct.es (C. Colodro-Conde), isidro.villo@upct.es (I. Villó-Pérez), jm.ferrandez@upct.es (J. Manuel Ferrández).

http://dx.doi.org/10.1016/j.neucom.2014.09.071
0925-2312/

emulate Carthago CNN architecture [5,6], demonstrating their viability and effectiveness for quickly prototyping of video processing applications based on CNNs.

Section 2 presents the proposed algorithm. Section 3 describes the system architecture and the software/hardware partitioning. Section 4 summarizes the main features of the CNN modules and details the organization and hardware implementation of the CNN architecture. Section 5 compares the results obtained when modeling the CNN architecture using two HLS tools. Section 6 shows the implementation and processing results which have been obtained with real data. Finally, Section 7 gathers main conclusions.

2. Algorithm to improve the spacial resolution of multi-stellar images

The main objective of the proposed algorithm is to minimize the adverse effect that atmospheric disturbance has on the images obtained by terrestrial telescopes. While in the space, the spatial resolution is just limited by the selected wavelength and the lent's diameter. On earth, however, atmospheric fluctuations produce variations on the diffraction index of the medium, what causes blurring of images. As a result, the quality of the images never achieves the ideal maximum angular resolution of the telescope i.e., the minimum angular distance that must be between two objects to be distinguished.

The Lucky Exposures Technique or Lucky-Imaging (LI), originally proposed by Fried in 1966 [7–9], is based on the idea of registering just the moments of maximum atmospheric stability. According to this consideration, if astronomical images (specklegrams) are acquired with short exposure time, using a high sensitivity, fast and enough low-noise sensor (such as EMCCD devices), some ones out of thousands of images, typically 5–15%, offer a much lower distortion than the others. If only these lucky images are taken into account and they are processed and recombined to reduce the information and to reach the desired sensibility, the resulting image can nearly reach the maximum resolution of the telescope, i.e., the resulting image offers a quality in the same order that it would offer if the image had been acquired beyond the atmosphere or in the absence of it. The difficulty here is that high performance processing is required, thus, keeping the adequate trade-off between quality of results and the computational cost of the algorithm is a major concern.

Several instruments have been developed based on LI principles, starting from the experiences of J.E. Baldwin's working group at Cambridge University [10–13], or later the FASTCAM instrument from the Instituto de Astrofísica de Canarias, in collaboration with the Universidad Politécnica de Cartagena [14–16].

FASTCAM combines an FPGA based system for high speed image transmission and evaluation, with proprietary software for the efficient processing of tens of thousands of images. Some of the main differences between the instruments reside in the reduction algorithm they include to combine the lucky images. The Shift And Add (SAA) [17] is the simplest and most used, and it has been demonstrated that SAA can reach the diffraction limit of the telescope under a wide range of conditions. It consists in shifting every image relatively to a reference star maximum and accumulating (adding) them. Images with greater peak values are considered less distorted and, therefore, lucky images. Modified SAA algorithms [18–20] have been proposed for specific conditions and applications. All these algorithms share the advantage of ranking among the least computationally intensive. In case of extended objects, or when there is not a reference maximum to re-center images, several methods have been proposed, based on the triple-correlation or its Fourier Transform, the bispectrum [21,22].

In a previous work [4], a platform for accelerating basic LI based algorithms was proposed. The processing flow included several CNN modules and implemented a modified *shift and add* registration algorithm using a single reference star. The system performed properly for images with a reduced number of punctual objects of interest. However, in case of wide field images (with tens of punctual objects), resolution enhancement decreased as the objects were farther from the reference star. Reaching the diffraction limit here requires more images, what means more observation and computation time. Another issue is related to the use of the maximum as the unique image quality factor. This might cause that highly noisy sparse speckles surpass the quality criteria.

In order to overcome the mentioned limitations, in this paper a new algorithm is proposed, which is well suited for multi-stellar images and improves resolution in wider zones of interest. Moreover, instead of considering just its brightness level, we propose to use also a form factor (Point Spread Function, PSF) indicator, measure of the speckle distortion, to determine the image quality. This algorithm, depicted in Fig. 1, requires however a greater computational effort, thus, it was necessary to redefine and optimize the architecture of the computing intensive stages. Fig. 2 shows the ideal result of each step of the proposed algorithm.

The proposed algorithm works with stacks of images (typically 1000 specklegrams) that are stored in the system memory. Based on the first specklegram, the user defines several regions where there is confined an object of interest and threshold values for the quality criteria (minimum brightness and PSF deviation) on each region. Then, a computationally intensive step processes every speckle independently and is followed by a low effort step where the resulting images are combined and post-processed to obtain the final image. The first step begins with a pre-processing stage that makes use of two CNN modules in parallel. CNN1 performs a band-pass filter of every speckle, fine tuned to the size of the desired objects. CNN2 implements a low-pass filter that will be used to estimate every speckle PSF. From these parameters and the threshold values selected by the user, the algorithm determines automatically if every region in an image has the minimum quality to contribute to the averaged region. If a speckle surpasses the quality threshold, the whole specklegram is re-centered with respect to the brightest pixel in that region and accumulated with the other specklegrams selected to contribute to the same region. The algorithm is repeated until the last image in the stack is processed. As a result, for every region, a full image is obtained which accumulates those specklegrams that had better quality in that region. In a second step the obtained images are combined to form a unique image of better quality. First, all images are re-centered again, relatively to the brightest pixel of the first region, and then, combined using a weighted average, in which the contribution of an image pixel is inversely proportional to its distance to the maximum of every region. Finally, the obtained image is post-processed by CNN3 module, which applies a smooth sharpening to highlight detected objects.

3. System architecture

The most immediate solution is to implement the algorithm with a software approach using off-the-shelf PCs. It is the fastest and simplest approach, but it presents the shortcoming of performance, since it requires too much time to emulate the CNN modules. A full hardware implementation can overcome this major drawback and achieve very fast execution times, but it implies a considerably bigger effort in the design process and much more development time [23]. We have adopted an intermediate approach, which combines the advantages of both