GPU based Metaheuristic for Fast Controller of an Evolutionary Robot

N. Benalia[1], N. Ouannes[1], N. Djedi[1], Y. Duthen[2]

1. Biskra University, Lesia Laboratory, BP 145 RP Biskra, Algeria
http://www.univ-biskra.dz/lab/Lesia/
2. UMR 5505 University, Vortex Team, IRIT Laboratory, Toulouse, France
http://www.irit.fr/-Equipe-VORTEX-

Keywords: Parallel Evolutionary Algorithms, Evolutionary Robotics, GPGPU.

1 Introduction

Bio-inspired techniques constitute a potential which continues to demonstrate its ability in solving complex problems. Nevertheless, due to their complexity and the execution's time they require, their use has been delayed for a long time in many domains. On the other hand, graphic modern cards play today an important role in the acceleration of sequential programs in various domains. Many algorithms are redesigned and rewritten for their execution on modern graphics cards (GPGPU).

Researchers in biomechanics, robotics, and computer science work to understand human natural motion in order to reproduce it in other forms. The objective of humanoid robotics' researchers is to obtain robots that can imitate the human behaviours to collaborate, in the best way, with humans. An obvious problem confronting humanoid robotics is the generation of stable and efficient gaits at a reasonable time. In order to address this problem, some biologically inspired control alternative methods have been proposed, which do not require the specification of reference trajectories.

Over the past decade, researchers have conducted investigations for parallelization of sequential applications built on the exploitation of evolutionary algorithms (EAs) and neural networks techniques [1], [2]. This paper constitutes a contribution towards this direction and presents a detailed study by investigating the effect of taking some parts of the physics simulator used on the GPU, chromosome size, population size, kernel configuration parameters on the speedup of the evolution of the robot controller. The key distinction of our approach is the effort to go beyond the chromosome level parallelism whenever possible and utilize the massively multithreaded model of GPUs to its fullest, taking into consideration the morphology of the robot and the environment of simulation.

The objective of this work is to propose a model that accelerates a technique we proposed in [3], which is the combination of an evolutionary algorithm and a recurrent neural network (RNN) that composes the brain of our robot. The choice of the RNN is based on the obvious advantage an RNN has over the traditional feed forward network is memory. The use of feedback connections allows the RNN to have a memory of past events. The main challenge of legged robots is the issue of creating controllers for them. The problem is difficult due to the number of freedom's degrees in each leg and the changes in the body centre of mass and momentum. In general, the designed controllers suffer from several problems for several reasons. The results of our study are compared to the serial algorithm by studying the effect of a number of parameters. It is significant to note that these are only comparisons of un-optimized GPU and CPU code. The performance can be further increased for both architectures.

The study we propose is ambitious within the constraints of the theoretical proposal. In all tests of this simulation, we have used a PC with 2 GTX 480 NVIDIA GPUs and find very promising results.

2 Proposal in Relation with Physic Simulation and Evolution on GPU

Over the past decade, physics-based simulation has become a key enabling technology for different applications. It has taken a front seat role in computer games, animation of virtual worlds and robotic simulation. Physics engines are considered one of the most important of a multitude of components requesting CPU time. Physics engines are used in a variety of domains. While some areas require high accuracy, others speed of simulation is more important. To attend this demand of acceleration, some physics engines are already using GPU acceleration like rigid body collision detection and particle interaction.

In our proposal, we adopted the GPGPU architecture proposed by Marcelo and al [4] in order to exploit the high accuracy of these kind of devices. The proposed architecture uses the GPU as a mathematics and physics coprocessor.

Since a Neural Network requires a considerable number of vector and matrix operations to obtain results, it is very suitable to be implemented in a parallel programming model and run on GPUs. Meanwhile, few studies have applied CUDA to neural networks as this field requires further investigations in applying the CUDA technology to neural computation and robotics research. The humanoid robot presented in this paper uses Elman Recurrent Neural Network for its biological plausibility and powerful memory capabilities. Furthermore, biological neural networks do not make use of back propagation for learning. Instead, we use evolutionary algorithms to evolve locomotive behaviours. In this work, we exploit the parallelism at a greater, so we form groups of threads to handle a single chromosome which corresponds to each robot. Moreover, the initialization of the first population is done in parallel, which means that all the RNNs are initialized and activated in parallel. We should mention here that an efficient mapping of the population is very important because the occupation rate of the graphics cards is a factor that directly affects the overall performance. Thus, mapping the problem to a massively multithreaded model for which GPUs are best suited. In this case, the reinforcement of good decisions relies on some feedback from the system itself and the exploitation of learned knowledge from the environment.

We are implemented our method using C/C++ and CUDA (4.0) and run experiments on PC which has an Intel i7 870 (2.93 GHz) processor and a two NVIDIA GeForce GTX480 GPU. The humanoid model used here is a fully tree-dimensional bipedal robot with 15 degrees of freedom. The robot model consists of 12 rigid-body parts, and 11 ODE joints. The performance of our proposed approach yielded a speedup ratio of 4.68-11.36 times compared to the CPU implementation method.

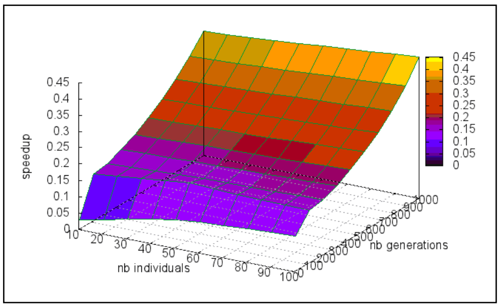

Figure 1. Global system execution speedup.

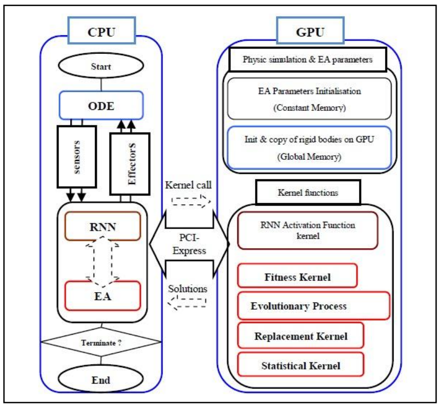

Figure 2. Configuration of RNN and EA with ODE on GPUs.

3 References

[1] Arora, A.; Tulshyan, R. ; Deb, K.: Parallelization of binary and real-coded genetic algorithms on GPU using CUDA, IEEE Congress on Evolutionary Computation, pp. 1-8, 2010.

[2] Peniak, M.; Morse, A. ; Larcombe, C. ; Ramirez-Contla, S. Aquila : An Open-Source GPU-Accelerated Toolkit for Cognitive and Neuro-Robotics Research. San Jose, CA July 31 2011-Aug. 5 2011.

[3] Ouannes, N., N. Djedi, N., H. Luga , H., & Duthen , Y. : Gait Evolution for Humanoid Robot in a Physically Simulated Environment. In Intelligent Computer Graphics 2011, Studies in Computational Intelligence, Dimitri Plemenos Georgios Miaoulis (Eds.), Springer pages157-175, 2012.

[4] Marcelo P. M. Zamith., Esteban W. G. Clua , Aura Conci , Anselmo Montenegro., Regina C. P.,Leal-Toledo., Paulo A,. Pagliosa, Luis Valente : A Game Loop Architecture for the GPU Used as a Math Coprocessor in Real-Time Applications. In Magazine Computers in Entertainment (CIE) - SPECIAL ISSUE: Media Arts, Vol.6, (2008)