Multi-Agent System with Multiple Group Modelling for Bird Flocking on GPU

R. Hidayat, D. Spataro, E. De Giorgio, W. Spataro, D. D'Ambrosio

Speaker:
Davide Spataro
Ph.D student at *Department of Mathematics and Computer Science@*
University of Calabria

STFCM
Scienze e Tecnologie Fisiche, Chimiche e dei materiali

PDP16 Heraklion Crete, Greece
February 18th, 2016

PDP 2016

Abstract

Birds flocking is an interesting natural phenomenon of investigation. In this paper, we present a GPGPU model for birds flocking simulation using NVIDIA's CUDA framework. This technology has been widely adopted in computational science and have increased computation performances. We present the ACIADDRI model for

- aggregate motion of a large number of birds in virtual environment;
- other species or predators avoidance in the plane

The work shows that the use of the CUDA technology can be effective to cut computational costs also in multi-agent modelling.

Contents

Introduction

Flocking behaviour exists in nature at almost every length scale of observation but it can be investigated as a collective motion.
One study approach is based on mathematical modelling and numerical simulations:

- random motion, organisms are treated like gas molecules their motion is Brownian combined with attraction/repulsion forces;
- ordinary differential equation (ODE) models.

However organisms seldom move really randomly, nor are they just simple particles.

$$\downarrow$$

Individual-based models or even multi-agent models seem a better choice.

ACIADDRI (*Aggregate Collection of Interactive Agents using nviDia's cuDa Reliable Informatics*) multi-agent flocking model extends Reynolds works on bird flocking behaviour by adding

- multi-species interaction,
- predator avoidance,
- partially observable environment and birds flight constraints.

We carried out experiments on the specific GPU hardware and by considering

- aggregate motion of a huge number (up to tens of millions) of boids in a virtual environment,
- other species or predators avoidance.

Significant performance improvement in terms of speedup were obtained (up to 500x).

The ACIADDRI model

Craig W. Reynolds in 1987 was amongst the first to abstract the flocking behaviour, in order to steer a swarm of simulated birds, called boids (birdoid).
Every boids has some limitations that can be specified by the three behavioural rules:

COHESION: to attempt to stay close to nearby flockmates;

COLLISION AVOIDANCE/SEPARATION:
to evade obstacles and flock mates which are too close;

VELOCITY/HEADING MATCHING:
also called *alignment*, to move in the same direction as nearby flock mates.

Model parameters:

In ACIADDRI model the environment and each bird's species is described by sets of parameters.

Environment parameters:

Name	Symbol	Dimension	Description
Length	W_x	[**L**]	Length of the environment
Height	W_y	[**L**]	Height of the environment
Width	W_z	[**L**]	Width of the environment
Time Step	t	[**T**]	Computational step duration

Table: List of Environment Parameters of Birds Flocking

1 pixel $\backsim$ 1 meter.

1 time step $\backsim$ 1 processing time.

Species parameters:

Each specie is described by a set of parameter that represent quantities that are involved in the flight and flocking dynamics.

- Bird's wingspan s is an approximation of the volume it occupies;
- v_p is the maximum velocity it can travel and a is bird's maximum acceleration.

Viewing frustum: FOV together with the maximum sight distance.

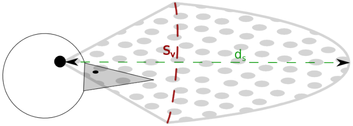

Figure: Birds vertical field of view.

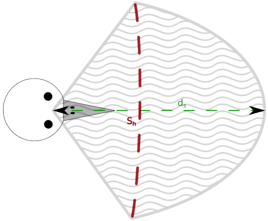

Figure: Birds horizontal field of view.

Bird's field of view:

Each bird has a limited visual capacity described by its field of view (FOV, $\mathscr{F}_p$). Let $\mathbf{p'_n} = \langle p_n^x - v_o^x, p_n^y - v_o^y, p_n^z - v_o^z \rangle$ the position vector of the object *n* in the *o*'s frame of reference, then *n* is *o*'s neighbour if and only if the followings hold:

$$\delta_s = ||\mathbf{p_o} - \mathbf{p_n}||, \ \delta_s \leq d_s$$

$$-\frac{s_h}{2} \leq \theta \leq \frac{s_h}{2}, \quad -\frac{s_v}{2} \leq \phi \leq \frac{s_v}{2}$$

where s_h is the maximum horizontal range of view, s_v is the maximum vertical range of view and

$$\phi = \arccos\left(\frac{p_n'^z}{\sqrt{(p_n'^x)^2 + (p_n'^y)^2 + (p_n'^z)^2}}\right), \quad \theta = atan2\left(\frac{p_n'^y}{p_n'^x}\right)$$

Cohesion:

In formal terms $\vec{C}_b^i$, the bird b's centroid at time i, is given by:

$$\vec{C}_b^i = \frac{1}{|\mathcal{N}_b|} \sum_{n=1}^{|\mathcal{N}_b|} \vec{p}_n \frac{d_{i,j}}{d_s}$$

where:

1. $\mathcal{N}_b$ the set of birds in b's FOV.
2. $\vec{p}_n$ is the position of $n \in \mathcal{N}_b$, set of neighbours
3. $d_{i,j}$ is the distance between bird b and its neighbour c

The b's cohesion vector $\vec{v}_c$ is then defined as follows.

$$\vec{v}_c^i = \begin{cases} \frac{\vec{C}_b^i - \vec{p}_b}{||\vec{C}_b^i - \vec{p}_b||} + a, & \text{if } 0 < |\vec{v}_d| \leq v_p \\ v_p, & \text{otherwise} \end{cases}$$

Separation: A bird try to keep certain distance between itself and its neighbours. Bird b's separation velocity $\vec{S}_b^i$ at time i is given by:

$$\vec{S}_b^i = \begin{cases} \left[\sum_{j \in \mathcal{N}_b} \frac{\vec{p}_b - \vec{p}_j}{||\vec{p}_b - \vec{p}_j||} \; f_s\right] + a, & \text{if } 0 < |\vec{S}_i| \leq v_p \\ v_p, & \text{otherwise} \end{cases}$$

where:

1. $\mathcal{N}_b$ is the set of neighbours,
2. $f_s = \begin{cases} 0 & \text{if } d_{i,j} > d_{min} \\ 1 - \frac{d_{i,j}}{d_{min}} & \text{otherwise} \end{cases}$

d_{min} is the minimum distance between two birds to avoid collision.

Alignment: Bird b's alignment $\vec{A}_b^i$ is here defined as

$$\vec{A}_i = \left[\sum_{j \in \mathcal{N}_b'} \vec{v}_j \, f_a \right] + a, \quad 0 < |\vec{A}_i| \leq v_p$$

where:

1. $\mathcal{N}_b' \subseteq \mathcal{N}_b$ is the set of birds considered by b for the alignment (e.g. the nearests).
2. $\vec{v}_j$ is the j's velocity.
3. f_a is the alignment coefficient. Let $d_{i,j}$ the distance between bird i and j then f_a is given by:

$$f_a = \begin{cases} 0 & \text{, if } d_{i,j} > d_a \\ 1 - \frac{d_{i,j}}{d_a} & \text{, otherwise} \end{cases}$$

d_a: maximum distance bird consider to align

Other species and predator avoidance: ACIADDRI is a multi-agent with multiple group model:

1. Other species avoidance

This behaviour is similar to *separation* with the difference that a only birds from other species are taken into consideration. In formal terms the other specie avoidance vector $(\vec{\tau}_i)$is given by:

$$\vec{\tau}_i = \left[\sum_{j \in \mathcal{N}_b} \frac{\vec{p}_b - \vec{p}_j}{||\vec{p}_b - \vec{p}_j||} \; f_s \right] + a$$

where:

1. $\mathcal{N}_b$ is the number of neighbours of specie different from the one of b,
2. $f_s = \begin{cases} 0 & \text{if } d_{i,j} > r_s \\ 1 - \frac{d_{i,j}}{r_s} & \text{otherwise} \end{cases}$

   r_s is the minimum distance bird avoid other species

2. Predator avoidance

The predator avoidance vector is computed by taking in consideration position and velocity (speed and heading) of all the predators within the bird's FOV. The predator avoidance vector $\vec{\Gamma}^i_b$ is defined as follows:

$$\vec{\Gamma}^i_b = \left[\sum_{j \in \mathscr{P}_b} \frac{\vec{p}_i - (\vec{p}_j + \vec{v}_j)}{||\vec{p}_i - (\vec{p}_j + \vec{v}_j)||} \; f_p \right] + a, \quad 0 < |\vec{\Gamma}_i| \leq v_p$$

where:

1. $\mathscr{P}_b$ is the *b*'s set of predators
2. $f_p = \begin{cases} 0 & \text{if } d_{i,j} > r_p \\ 1 - \frac{d_{i,j}}{r_p} & \text{otherwise} \end{cases}$
   is the predator avoidance coefficient, where r_p is the minimum distance bird avoid predator.

Wandering:

When the neighbourhood of a bird is empty it flies pseudo-randomly in the space. This kind of behaviour is called *wandering*. Wandering is obtained combining a current and a random direction

$$\vec{\omega}_b^i = \begin{cases} 0 & \text{if } \mathcal{N}_b \neq \emptyset \\ \frac{\vec{s}}{|\vec{s}|} \, w_r + w_d, \; 0 < |\vec{\omega}_i| \leq v_p & \text{otherwise} \end{cases}$$

Where

- $\vec{s}$ is the size of the bird;
- w_d is the maximum wandering distance;
- w_r is the maximum radius wandering from the target;
- v_p is the maximum velocity.

Flight model:

Bird *b* flight at step *i* is described by

$$\vec{p_b}^{\,i} = < p_x^i, p_y^i, p_z^i >, \quad \vec{v_b}^{\,i} = < v_x^i, v_y^i, v_z^i >$$

The evolution of the bird's velocity over time is regulated by

$$\vec{v}_b^{i+1} = (rsin\theta' cos\phi',\ rsin\theta' sin\phi',\ rcos\theta')$$

where:

- $\theta' = \begin{cases} \theta_d & \text{if } |\theta_b - \theta_d| < \theta_{max} \\ \theta_b + \theta_{max} & \text{otherwise} \end{cases}$
- $\phi' = \begin{cases} \phi_d & \text{if } |\phi_b - \phi_d| < \phi_{max} \\ \phi_b + \phi_{max} & \text{otherwise} \end{cases}$

The polar angle θ_d and the azimuth angle ϕ_d are associated to

$$\vec{v}_d^i = \mu_c \vec{v}_c + \mu_s \vec{v}_s + \mu_a \vec{v}_a + \mu_A(\vec{\tau}_i + \vec{\Gamma}_i) + \vec{\omega}_i$$

where

- μ_c, μ_s, μ_a are the cohesion, separation and alignment coefficient (social coefficients) and μ_A is the avoidance coefficient,
- $\vec{v}_c^i$,$\vec{v}_s^i$, $\vec{v}_a^i$ are the *social velocities*:
  - $\vec{v}_c^i$, the cohesion velocity,
  - $\vec{v}_s^i$, the separation velocity,
  - $\vec{v}_c^i$, the align velocity.
- θ_d, θ_b are the polar angle of the velocity vector $\vec{v}_b^i$ and $\vec{v}_d^i$ respectively.
- ϕ_d, ϕ_b are the azimuthal angle of the velocity vector $\vec{v}_b^i$ and $\vec{v}_d^i$ respectively.
- $\vec{\omega}_i$ is the wondering vector.

GPGPU parallel implementation

In this work we adopt GPUs and the CUDA framework to accelerate the flocking simulation of a large number of boids using the model presented in an environment with a number of agents up to $5x10^6$. We produced two different parallel versions consisting in (the well-known host-managed accelerated program structure):

- Initialization of data structures on *CPU*
- Data transfer from ***CPU* to *GPU***
- Kernels execution on *GPU*
- Copying the result back from ***GPU* to *CPU***

The parallelization strategy is designed with the purpose to avoid as much as possible the very undesirable data copy from *host to device*, or vice versa.

- The computation of $\vec{p}_b^{i+1}$ and $\vec{v}_b^{i+1}$ is entirely performed on GPU and implemented as composition of CUDA kernels.
- Parameters are stored in constant memory for fast access.
- An OpenGL 3D visualization tool comes with the simulation system and permits real time and interactive rendering of the flocking model.

Naïve version:

Each agent is mapped to a CUDA thread organized in a 1D block-grid structure.

- All data resides in global memory and user managed cache (shared memory) is unused.
- This version already gives rise to a speedup of $\approx 20x$.
- An optimization was carried out by considering the *If-Divergence mitigation*
- The serialization presents a performance loss.

Shared memory version:

The shared memory is exploited in order to cache bird's frequently accessed data.

- Shared memory is much faster than global memory,
- although it is of limited capacity (and depends on compute capability of the device), only accessible at block level and cleared at each kernel invocation.
- The adopted strategy divides the computation in a number of phases that depend on the chosen block size. Each phase can be then performed exploiting the fast memory.

Different tests were carried out by considering different number of boids and an environment composed of $1000 \times 1000 \times 1000$ cells. Each simulation was carried out for 10^4 time steps.

Three devices were adopted for testing different CUDA version of the model: the high-end GTX 980, a GT 635M and a low-end mobile chip. Above are shown the execution times of the naïve and shared memory version:

# birds	Sequential	GT 635M (×)	GTX 980 (×)
1024	263.9	29.1, (9, 07)	10.5, (25, 13)
5120	4913.0	574.4, (8, 55)	51.7, (95, 02)
10240	19074.5	2241.6, (8.51)	109.0, (174, 99)
15360	43332.3	5004.7, (8.65)	235.2, (184, 23)
20480	86065.7	8868.9, (9, 70)	312.5, (275.408)
40960	452423.1	-	1023.8, (441.90)
81920	1966134.9	-	3663.5, (536, 70)
163840	8003173.0	-	14877.4, (537, 94)
327680	35815012.0	-	58003.0, (617, 47)

Table: Timing (in seconds) for the Parallel CUDA Naïve implementation

Timing (in seconds) for the Parallel CUDA shared memory implementation:

# birds	Sequential	GT 635M	GTX 980
1024	263.9	19.9	7.9
5120	4913.0	366.3	34.6
10240	19074.5	1398.7	96.5
15360	43332.3	3110.0	154.9
20480	86065.7	5522.0	280.8
40960	452423.1	-	825.4
81920	1966134.9	-	3307.2 -
163840	8003173.0	-	13565.9
327680	35815012.0	-	54113.4

We see an evident cut down of computational time by GPGPU use.

- Starting from Reynolds's behavioural model, we have presented a preliminary multi-agent and multiple group approach for bird flocking coupled with an efficient implementation by means of the CUDA framework.
- We have proven the full suitability of the GPGPU paradigm for efficiently simulating multi-agent systems.
- Future developments:
  - improve the bird's light modelling to better adhere with aerodynamics theory;
  - adding the possibility of the environment to contain obstacles, and the usage of multi-gpu hardware.

Thanks for the attention