Simulation of complex systems

The goal of this project is to develop a parallel tools to simulate the behavior of a group of particles. You will have to write a code that perform the simulation and then the corresponding parallel code.

Your code should be optimized and commented. You may work in team of two.

1 Flocking birds and schooling fishes

In this section, we consider the motion of a variety of particles that follows a simple set of rules. We will refer to this motion as the flocking behavior.

1.1 Introduction

Flocking is a generic term that describe the motion of a group of agents that follows a motion strategy designed to hold the group together. Such behavior is exemplified by the motion of groups animals, such as birds, fish, insects, and other types of herding animals.

In flocking it is assumed that the all the agents are following the exact same rules and apply the exactly the same algorithm. The only difference between the agents are their relative positions. Even with simple rules complex formations can be observed.

The spontaneous generation of complex behavior from the simple actions of a large collection of dynamic entities is called emergent behavior. As we will see in the next section, the algorithms that describes flocking behavior do not involve very sophisticated models of intelligence. However, even simple variants of this method can be applied to simple forms of crowd motion in games or motion pictures.

1.2 A simple model: boids

C. W. Reynolds wrote one of the most used model for flocking behavior in 1986 with his work on “boids”. In this system, the behavior of each agent is regulated by the combination of four simple rules. In order to give a mathematical description to the motion of boids, each boid is assigned a position $x \in \mathbb{R}^3$, a velocity $u \in \mathbb{R}^3$ and a mass M.

1. Separation: Each boid tries to avoid collisions with other nearby boids. Each boid emits a repulsive potential field whose radius of influence extends to its immediate neighborhood. Whenever another boid gets too close, the force from this field will tend to push them apart. Each boids try to move away from the local center of mass of the group. The rule writes

$$u_i = u_i - \frac{w_s}{M} \sum_{j=0, j\neq i}^{N} (||x_j - x_i||_2 < r_s) \cdot (x_j - x_i) \tag{1}$$

   where w_s is the weight of the separation force and r_s its the radius.

2. Alignment: Each boid's direction is aligned with nearby boids. Thus, local clusters of boids will tend to point in the same direction. The rule writes

$$u_i = u_i + \frac{w_a}{M} \sum_{j=0, j\neq i}^{N} (||x_j - x_i||_2 < r_a) \cdot u_j \tag{2}$$

   where w_a is the weight of the separation force and r_a its the radius.

3. Cohesion: Each boids tend to be drawn towards the center of mass of the flock. Of course, the boid cannot know where the center of mass is exactly since it is implicit and hence, the center of mass cannot be computed exactly. The rule writes

$$u_i = u_i + \frac{w_c}{M} \sum_{j=0, j\neq i}^{N} (||x_j - x_i||_2 < r_c) \cdot \cdot (x_j - x_i) \tag{3}$$

   where w_c is the weight of the cohesion force and r_c its the radius.

4. Avoidance: Each boid will try to avoid the collision with obstacles. In order to simulate avoidance, the obstacle will generate a repulsive potential field. With such a field, as a bold approaches the object, this repulsive field will deflect the boid from its flight path, thus avoiding a collision. Avoidance can also be applied to predators.

1.3 Implementation

In order to advance the position of the boids, we need to solve an initial value problem for each particle

$$\frac{x}{t} = f(x, t) \tag{4}$$

In the forces mentioned in the previous section, the integration was done using Euler method, but higher order methods can be used in order to improve the stability of the simulation, especially for large a number of particles.

For the efficiency reasons, we may not want to compute all the interactions between all the particles since it will take $O(n^2)$ time for each time iteration. In order to avoid this cost, the particles are stored in a grid or a quadtree (octree in 3D).

One issue that arises with this or any dynamical system is how to avoid undesirable (meaning unnatural looking) motion, such as collisions, oscillations, and unrealistically high accelerations. Here are two approaches:

2 Required tasks

The work consists of several tasks.

2.1 A first implementation

The first task is to implement a flocking simulation, in sequential, with a stable integration scheme, using the particle simulation and the four flocking forces: separation, alignment, cohesion, and obstacle avoidance.

You should think in advance what are the data structures and code organization will allow good parallelism.

2.2 Parallelism

The next step is to write a parallel version of the code using either shared memory or distributed memory paradigm.

3 Evaluation

You work will be graded on the following aspects

1. Quality of your program.
2. Performances of your application.
3. Analysis and relevance of your results.
4. Initiative
5. Quality of your report.

You report should be in pdf format and your code commented.
[]

References

[1] Craig W. Reynolds. Flocks, herds and schools: A distributed behavioral model. *SIGGRAPH Comput. Graph.*, 21(4):25–34, August 1987.