A Flocking Boids Simulation and Optimization Structure for Mobile Multicore Architectures

Mark Joselli     Erick Baptista Passos     Jose Ricardo Silva Junior
Eduardo Soluri     Marcelo Zamith     Esteban Clua.

Neighborhood Grid

Most of the research on flocking boids simulations tries to avoid the high complexity of proximity queries by applying some form of spatial subdivision to the environment and closed boid entities among the cells based on their position. Since the Renderscript is very new and have some scatter constraints, there are lack of spatial subdivision techniques implemented in this technology, so most works uses the brute force algorithm, which have a quadratic complexity. In this paper, instead of using a similar approach, we propose a novel simulation architecture that maintains entities into another kind of proximity based data structure, which we call Neighborhood Grid (NG). This data structure, each cell must fit only one entity and does not directly represent a discrete spatial subdivision. The Neighborhood Grid is an approximate representation of the system of neighborhoods on the environment, which maps the N-dimensional environment to a discrete map (lattice) with N dimensions, so that entities that are close in a neighborhood sense, appear close to each other in the map. Another approach is to think of it as a multi-dimensional compression of the environment that still keeps the original position information of all entities.

Tests and Results

For the purpose of this work, we choose to validate the proposed technique by implementing a well known distributed simulation algorithm called, flocking boids. This is a good algorithm to use because of its good visual results, proximity to real world behavior observation of animals and understandability. The implementation of the flocking boids model using our algorithm enables a real-time simulation of up to thirty thousand animals of several species, with a corresponding visual feedback.

Introduction

In a typical natural environment, it is common to a huge number of animals in a densely populated systems or densely populated systems, such as sport arenas, communities of ants, bees and other insects, or even streams of blood cells in our circulatory system. Computer simulations of these systems usually present a very limited number of independent entities, mostly with very predictable behavior. Algorithms for flocking simulation are driven by the need to avoid the squared complexity of the proximity queries between entities. Several approaches have been proposed to cope with this issue but none of them has reached an ideal level of scalability. As far as we know, no work until the present date has proposed a real time simulation of more than just a few hundreds of complex entities interacting with each other on a mobile device.

Google introduced the Honeycomb version of Android the Renderscript API (application programming interface). Renderscript is an API for achieving better performance on Android phones and tablets. Using this API, applications can use the same code to run on different hardware architectures like different CPUs (Central Processing Units), ARM (Advanced RISC Machine) v5, ARM v7, and X86 CPUs (Graphic Processing Unit and DSPs (Digital Signal Processors)). The API decides which processor will run the code on the device at runtime, choosing the best processor for the available code. This work presents a novel modeling of flocking boids data structure suitable for this new architecture and compares it to the traditional brute force algorithm. As far as the authors know, this is the first flocking boid simulation that uses this kind of approach.

Mobile GPU

In the past few years, mobile phones and other mobile devices have gone from simple phones to messaging devices to high end smartphones with serious computing capabilities. Nowadays, most of these devices are equipped with multicore processors like dual-core CPUs and GPUs, which are designed for both low power consumption and high performance computation. Moreover, most devices still lack libraries for generic multicores computing usage, such as CUDA or OpenCL. However, computing certain specific kind of tasks in these mobile GPUs, and other available multicores processors, may be faster and much more efficient than their single threaded CPU counterpart.

In this work, we present a novel approach for flocking boids simulation based on the Android renderscript API. We describe and implement a custom NGrid, and present results with a simple game based on this platform.