Real-time Simulation of Smoke through Parallelizations

Torbjørn Vik, Anne C. Elster and Torbjørn Hallgren

Norwegian Univ. of Science & Technology (NTNU), Norway

Computer simulations of natural phenomena such as clouds, water and fire have for a long time been an area of intensive research. These fluid-like motions have extremely complex behavior, and their chaotic and unpredictable behavior render them into fascinating visual effects. Even though the equations describing these effects have been known for over a century, their complexity makes them difficult and computationally expensive to imitate and visualize accurately. Efficiently solving these equations, while still maintaining sufficient visual quality, has been a goal of researchers for decades.

This paper describes how to generate visually correct 3D real-time smoke images where the audience does not recognize significant errors compared to their real-world experiences; i.e., a credible visual simulation. The result is a 3D parallel implementation producing more than 20 frames per second on today's modern desktop computers.

The simulation model used is based on the assumption that smoke can be modeled as an inviscid, incompressible fluid, using Euler's equations. The algorithms are largely based on the ones presented by Fedkiw, Stam and Jensen in their paper entitled *Visual Simulation of Smoke*, and are analyzed with respect to efficiency and visual quality.

All the algorithms involved are discussed from a parallel programming point of view, and two different parallel SMP implementations are presented. More specifically, Intel based SMP systems running MS Windows were used as test-beds.

We introduce techniques for dynamic job distribution and pipelining, thus reducing the penalty caused by serial sub-tasks such as rendering. The technique of pipelining is well-known, especially within computer hardware engineering, but this work shows its relevance within the field of visualization. And as most of today's SMP systems still share a common graphics card, the techniques could apply to a wide range of visualization software. Results showing that certain parts of the algorithm gained significant speedups by using Intel's SSE instructions, especially the Conjugate Gradients method, are also presented.