
Design and Implementation of a Generic Framework for Virtual Reality Physics Laboratory

Fatmah S. Balamash
KING ABDULAZIZ
UNIVERSITY

Mahmoud I. Kamel
KING ABDULAZIZ
UNIVERSITY

Mohamed Y. Dahab
KING ABDULAZIZ
UNIVERSITY

Fouad Al-Dehlawi
KING ABDULAZIZ
UNIVERSITY

ABSTRACT

Over the past decade, educational software has made significant evolution by providing the means for interactive applications. There are, however, various categories of applications such as virtual laboratories, where these applications show some difficulties, such as ad-hoc design rather than generic design, to be developed. In this paper, we design and implement a generic application for virtual reality physics laboratory. The framework does not only support developing the well-known physics experiments, but also it supports creating and inventing new experiments. The main idea behind this application is to model particle as a generic concept that can enhance building several experiments in virtual reality physics laboratory. The design and implementation of this application program leads to produce libraries of particles that can be easily modified, extended, and used in simulating different physics experiments. To evaluate the given proposal, three different experiments have been successfully developed.

Keywords

particle systems, software reusability, virtual reality, simulation, visualization.

INTRODUCTION

Experience has shown for many universities and scientific research centers, the importance

of virtual laboratories in education especially in physics because experimentation is the best way to help learners to study physics, instead of passive learning. In absence of capabilities and possibilities, the virtual laboratory is a feasible alternative for conducting scientific experiments. Several researches and efforts have been done in virtual laboratories to allow learners to understand complex systems and observe their behavior. Virtual laboratories show some difficulties to be developed especially physics virtual reality laboratories because physics science consists of many complex processes and these processes are hard to understand. In this paper we propose a generic framework to enhance the development of virtual reality physics laboratory which supports creating the well-known physics experiments in addition to support creating new experiments. The main idea behind this framework is to model particle as a generic concept that can support building several experiments in virtual reality physics laboratory. The framework depends heavily on the idea of particle system which was introduced by [13] to model fuzzy objects such as fire, clouds and water. We specify algorithms for implementing the model on a computer using object oriented approach. The computer program can simulate the physical system and define the computer experiment.

Such a computer experiment serves as a bridge between laboratory experiments and theoretical calculations. The remainder of this paper is structured as follows: Next section describes briefly the particle systems. In section 3, we present a literature review for virtual reality (VR) laboratories that apply the idea of particle systems. Section 4 describes the application of virtual reality of physics laboratory and an experiment. . In section 5, we evaluate the framework by three different test cases. Finally, section 6 is devoted to conclusions and future work.

Particle Systems

The term particle system has been used to describe modeling techniques, rendering techniques, and even types of animation. Particle systems are a collection of discrete entities called particles. Each particle is described by its state and a collection of attributes that determine its appearance and other factors. Particle systems used to model natural phenomena such as fire and water, snow falling, and the behavior of crowds. Important examples of this are the motion of atoms or molecules in solids and liquids, ions in hot gas plasmas, and electrons in semiconductor devices.

Particle systems simulations are widely used in physics [10], [13], [5] and [14]. Many experiments can be developed using particle systems such as free fall simulation, projectile motion, circular motion, two particles connected by a cord, molecular dynamics simulation of hard spheres and etc. What is common in all these is the use of a system of particles whose behavior is controlled by a set of rules. Each object in particle system has set of attributes which includes the following attributes [13]:

- Position
- Velocity (speed and direction)
- Color
- Age

- Shape (particle may be of any shape [17])
- Size
- Transparency

In addition to motion particles can make forcing, damping, colliding, binding and attracting with other particles. Also, each particle goes through three distinct phases in the particle system: generation, dynamics, and death.

Related Work

Particle systems have been used successfully for decades in computer science [2], [7], [8], [10], [19] and [6] and in physics [17], [3] [1] and [16] to model complex dynamics. They are used for simulating non-linear dynamics of complex systems that are hard or impossible to implement. Formally, particle systems capture so called basic emergence [9], which means behavior that is reducible to particle-to-particle interactions without any evolutionary processes involved. Particle systems are typically used for simulating fluids and gasses as well as elastic and deformable bodies [17], [3] and [16]. Particle systems are computationally attractive, because the models are simple difference equations over matrices independently of the underlying complexity of the simulated objects. Consequently, the models are compact and computationally lightweight. Particle systems can be seen as a complementary approach to model with finite element methods [4] and [6]. When using a finite element method, the space is partitioned into a predefined mesh and the focus is on the flow of matter and forces within the mesh, whereas with particle systems, the focus is on individual particles and their interaction within a continuous Euclidean space. Particle systems can be seen as also a complementary approach to use of spring-mass models [15], [12] and [18]. When using a spring-mass model, the focus is in defining the springs and their coefficients between the mass points which represent rigid undeformable bodies. In

contrast to this, when using particle systems, the interaction properties are assigned to the particles and particle-to-particle forces; in particular, the particle-to-particle forces may well be temporal and they do not need to be spring-like forces at all.

The Generic Virtual Reality Physics Laboratory Framework

We pretend that all physical phenomenon can be implemented as particle systems and we propose, in this section, an extensible robust framework for particle systems. The framework has mainly three basic classes namely: Particle, Particle Systems and Force. The relation and interaction among these classes has been depicted in figure (1). This framework is extendable by adding different classes and interact them with the basic classes, as we will explain later.

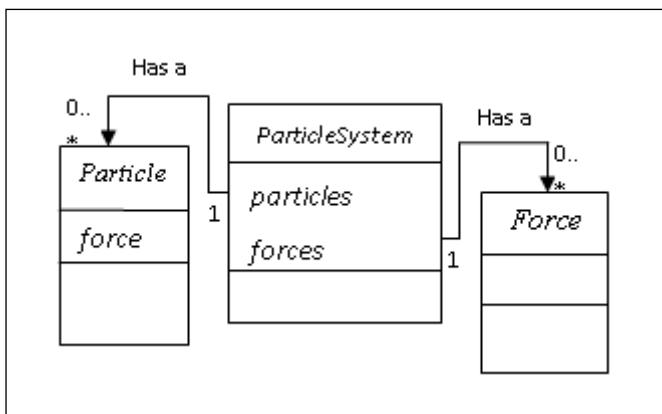


Figure 1: Particle system model

Particle Systems Class

ParticleSystem class has a set of properties. The most important properties include: *particles*, *forces*, and *vworld_handle*. The *particles* property is a set of instances, objects, of *Particle* class. The *forces* property is a set of instances, of *Force* class. The *forces* property contains all forces affected on the *particles*. *ParticleSystem* is the base class in the program. *ParticleSystem* class is not only handles and manages a list of *particles* and

forces but it also includes *vworld_handle* which handles the virtual world with GUI methods.

ParticleSystem class also has a set of methods. The most important methods are: *update_position()*, *update_force()*, *update_velocity()*, *update_attraction()* and *draw_particle()*. We implemented a *ParticleSystem* and use it to simulate some test case based on the Particle-Particle method by looping the following steps:

1. Accumulate the forces on each particle
2. Step forward one time step using a standard differential equation solver.
3. If desired, render the particle system using the particles attributes.

The first two steps are routine in any simulation using models whose evolution can be described by differential equations. The third step is a key to the graphical and visualization applications of *ParticleSystem*. Here we use the state of the *ParticleSystem* to determine an image.

Particle Class

As mentioned before, a *ParticleSystem* has a collection of independent objects called *particles*. Each element in *particles* represents an instance from a class called *Particle*. *Particle* has a set of properties such as *mass*, *charge*, *position*, *velocity*, *force*, *size*, *shape*, and *particle_handle*.

Force Class

Force class has a set of properties such as *attractions*, *springs*, *tensions* and *frictions*. Also, this class has a set of methods to accumulate and update each force type.

Virtual Reality Modeling Language (VRML)

The Virtual Reality Modeling Language (VRML) represents a very simple implementation of Virtual Reality (VR) using MATLAB. Virtual reality viewers handle all the computation for the 3D using a set of library methods. These libraries include methods for shading, texture mapping, texture

filtering, anti-aliasing, lighting, geometry transformations, and ..etc [11].

Evaluation

Our system is an extensible system. It is able to operate on the default particle structure and allowing at the same time to augment additional properties by using subclasses from the basic classes.

First test case (Molecule Dynamic Simulation)

To develop the well known Molecule Dynamic Simulation, we define a new object called Molecule particle that adds a new property called *CollisionCount*. By inheriting from Particle. Others new classes add like following: *Event* class to store collision event between particles and will then inserted onto the *PriorityQueue* class. Figure (2) depicts the extendible Particle system.

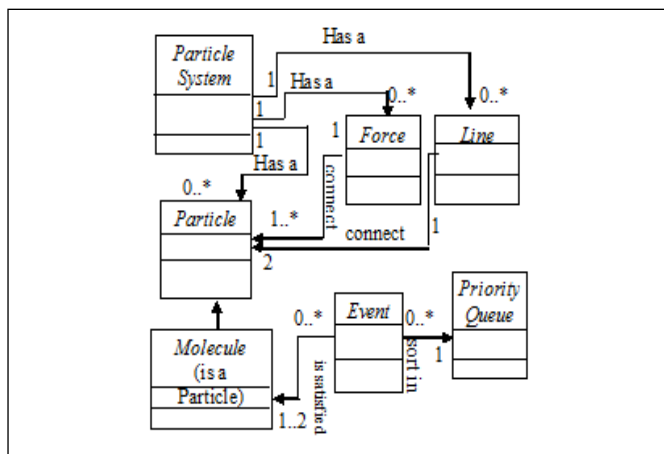


Figure 2: The extendible particle system class diagram

In our molecule dynamic (MD) experiment figure (3), we measure the kinetic energy, velocity root mean square, and collision frequency of the system at each simulated time step. We present the results of our MD simulations at different three temperature (173 K, 300 K and 370 K). To study the movement of a group of molecules and to compare the difference in the movement of each phase, we solved this problem in three examples in

table (1), and three examples in table (2). These tables show a performance for the object-oriented MD particle system. Comparing the experiments, we see the kinetic energy after the collision is always equal to the initial kinetic energy. From conservation of linear momentum, in the elastic collision (if there are no outside forces), kinetic energy was conserved, and also root mean square velocity was conserved.

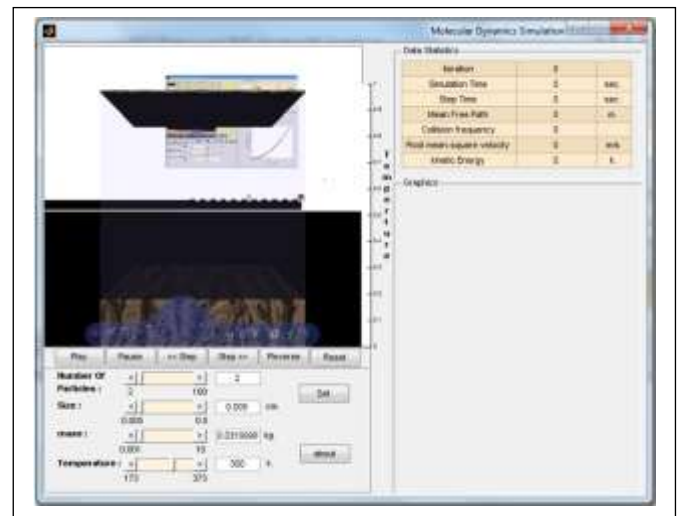


Figure 3: MD Simulation GUI

Second test case (Two particles Connected by a Cord)

As shown in Figure 4, Newton's second law of motion system, the simulation of two connected particles when friction is present. The gravitational acceleration is 9.81 m/s^2 was presupposed. The mass M_1 of the sliding particle, the value of the hanging particle mass M_2 and the coefficient of friction μ (within certain limits) can be changed. During the movement a red point in the time-displacement diagram indicates the present time and the covered distance and visualize the motion as it occurred. Simulates the motion of particles by solving Newton's Laws of motion using midpoint method. As soon as the measurement of time is finished, the measured

values will be displayed in the "Data Statistics" table.

Third test case (Free Fall)

This problem categorized as a particle moving in two dimensions with constant acceleration in the vertical direction. In this simulation, we can examine the motion of an object in free fall or rise, with or without air resistance and visualize the motion as it occurred. This is one-dimensional motion (vertical motion) under the influence of gravity, or drag force if there is air resistance. For example, the simulation drops both balls from rest from a height of 3 m, with a gravitational acceleration of 9.8 m/s/s directed down, with absent or present air resistance, and we should provide the initial position as well as the initial velocity of the object, see the initial condition in the table (3) and figure (5). We use the midpoint method to follow the motion of a free fall, of two balls with different attributes.

variety of motions, ranging from the motion of satellites in orbit to the motion of electrons in a uniform electric field.

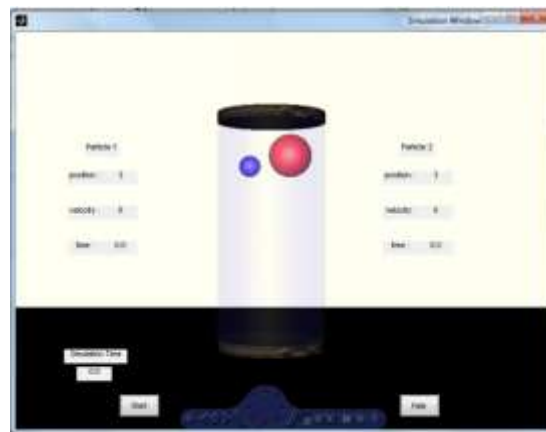


Figure 5: Free Fall GUI

There is still much work can be done to develop particle system. First, we will extend particle system to cover charged particle and light, lenses, mirror, and other physical system. The power of mathematics when applied to physics comes in part from the fact that frequently seemingly unrelated problems can have the same mathematical formulation. Hence, if we can solve one problem, we can solve other problems that might appear to be unrelated.

Acknowledgement

This paper was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, KSA. The authors, therefore, acknowledge with thanks DSR technical and financial support.

References

- [1] Azevedo, Vinicius C., and Manuel M. Oliveira. "Efficient Smoke Simulation on Curvilinear Grids." In Computer Graphics Forum, vol. 32, no. 7, pp. 235-244. 2013.
- [2] C. Foias, D.D. Holm and E.S. Titi, The navier-stokes-alpha model of fluid turbulence, Physica D: Nonlinear Phenomena 152153, pp. 505519, 2001.

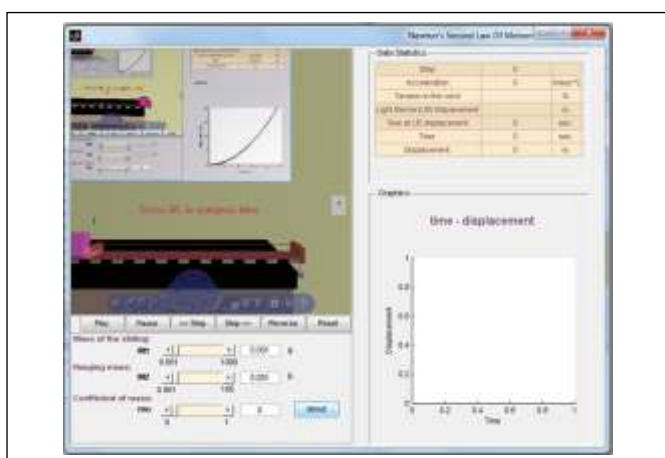


Figure 4: Newton2OfMotion GUI

Conclusions and Future Work

In this paper, we implemented a particle system and use this system to simulate some test cases. We have proposed a generic framework for physical system using object oriented approach. In this paper we explored the moving of a particle in three dimensions. This motion will allow us to examine a wide

-
- [3] N. Foster and R. Fedkiw, Practical animation of liquids, Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, ACM Press, New York, pp. 2330, 2001.
- [4] Goldstein, Rhys, and Gabriel Wainer. "Impulse-based dynamic simulation of deformable biological structures." In Transactions on computational systems biology XIII, pp. 39-60. Springer Berlin Heidelberg, 2011.
- [5] R.W. Hockney and J.W. Eastwood. Computer Simulation Using Particles, Adam Hilger, New York, 1988.
- [6] A. Ibrahimbegovi and D. Markovi, Strong coupling methods in multi-phase and multi-scale modeling of inelastic behavior of heterogeneous structures, Computer Methods in Applied Mechanics and Engineering 192,pp. 30893107, 2003.
- [7] J. Jansson and J.S.M. Vergeest, A discrete mechanics model for deformable bodies, Computer-Aided Design 34, pp. 913928, 2002.
- [8] H. Jin and J. Glimm, "Verification and validation for turbulent mixing", Nonlinear Analysis: Theory, Methods and Applications 69, pp. 874879, 2008.
- [9] A. Kubk, Toward a formalization of emergence, Artificial Life 9, pp. 4165, 2003.
- [10] V.S. Kumar, B.R. Kumar and P. Das, Web-spline based mesh-free finite element analysis for the approximation of the stationary navier-stokes problem, Nonlinear Analysis: Theory, Methods and Applications 68,pp. 32663282, 2008.
- [11] MathWorks, V.R., Virtual Reality Tool Box User's Guide, the MathWorks, Inc., Natick, MA, USA, 2002.
- [12] X. Provot, Deformation constraints in a mass-spring model to describe rigid cloth behavior, in: Proceedings Graphics Interface, pp. 147154, 1995.
- [13] Reeves W.T., Particle systems a technique for modeling a class of fuzzy objects, ACM Transactions on Graphics (TOG), Vol. 2, No. 2, pp. 91-108, 1983.
- [14] C. W. Reynolds, Flocks, Herds, and Schools: A Distributed Behavioral Model, Computer Graphics, vol. 21, no. 4, pp 25-34, 1987.
- [15] D. Terzopoulos, J. Platt, A. Barr, K. Fleischer, Elastically deformable models, ACM SIGGRAPH Computer Graphics 21, 1987.
- [16] S.T. Thornton and J.B. Marion, Classical Dynamics of Particles and Systems, (fifth ed.), Brooks/Cole Thomson Learning, Belmont, 2004.
- [17] D. Tonnesen, Dynamically coupled particle systems for geometric modeling, reconstruction, and animation, Ph.D. thesis, University of Toronto, Canada, 1998.
- [18] X. Tu, D. Terzopoulos, Artificial shes: physics, locomotion, perception, behavior, in: Proceedings of 21st Annual Conference on Computer Graphics and Interactive Techniques, ACM Press, pp. 4350, 1994.
- [19] J. Zhang, E. Angel, P. Alsing, and D. Munich, An Object-Oriented Particle System for Simulation and Visualization, XXVII Conferencia Latinamericana deInformat ica (CLEI 2001), September, 2001.

Table 1: The Results of the First Experiment

Temp-erature	No Of particles	Size	mass	Iteration	Collision frequency	V_{rms}	Kinetic Energy
173	11	0.01	0.032	30	1.5459e-9	4.8241e-10	4.0958e-20
				70	1.8006e-9	4.8241e-10	4.0958e-20
				100	2.9417e-9	4.8241e-10	4.0958e-20
300	11	0.01	0.032	30	2.3238e-9	6.3526e-10	7.1026e-20
				70	2.1762e-9	6.3526e-10	7.1026e-20
				100	3.0236e-9	6.3526e-10	7.1026e-20
373	11	0.01	0.032	30	2.6702e-9	7.0835e-10	8.8309e-20
				70	2.4199e-9	7.0835e-10	8.8309e-20
				100	4.0486e-9	7.0835e-10	8.8309e-20

Table 2: The Results of the Second Experiment

Temp-erature	No Of particles	Size	mass	Iteration	Collision frequency	V_{rms}	Kinetic Energy
173	100	0.02	0.032	100	3.5439e-7	3.8413e-10	2.3609e-19
300	100	0.02	0.032	100	4.6668e-7	5.0584e-10	4.0940e-19
373	100	0.02	0.032	100	5.2431e-7	5.6504e-10	5.1083e-19

Table 3: Initial Condition

Initial Condition				
properties	Ball1		Ball2	
Mass	10	kg	1	kg
Size(radius)	.4	m	.2	m
Initial_Position	[0,3.4,0]	m	[0,3.2,0]	m
Initial_velocity	[0,0,0]	m/s	[0,0,0]	m/s