Literature Review

Ziyan Xiong

October 5, 2021

Abstract

This literature review aims to review previous work in the filed of physically based simulation for modeling shapes and their motion. The first part of this review gives some intuitive examples on why this kind of simulation is useful. The second part introduces prior fundamental work for achieving different effects in physically based simulation. In the last part, this review will show some recent works in this filed and discuss how these application help to solve real problems in the production process.

1 Introduction

In the production of computer animation, it usually requires some skilled animators to implement realistic models and the movements of characters. Physically-based modeling provide an possible solution to synthesize complex shapes automatically. The physical laws of the simulation control the behavior of the models. Then animators can use this physics-based control system to guide their models. According to [TPBF87], the models are manipulated based on principles of mathematical physics. They react to applied forces (such as gravity), to constraints (such as linkages), to ambient media (such as viscous fluids), or to impenetrable obstacles (such as supporting surfaces) as one would expect real, physical objects to react. In this paper, authors develop elastically deformable models by constructing differential equations for the surface behavior under different physics effect based on simplified elasticity theory. The motion of the surfaces are obtained during numerically solving these differential equations. With these elastically deformable models, animators will no longer spend huge amount effort to create elastic surfaces such as cloth, plasticine, and skin and their motions. Moreover, the automatically generated motions are more accurate than the result from manual operation since they will always follow the physics laws in real world. The authors keep developing algorithms for Physically-based modeling and present the models for inelastic objects in [TF88]. In this paper, authors expand the motions of objects to a more realistic form, the simulated surfaces have their own inelastic properties. These newer dynamic models can tractably simulate three canonical inelastic behaviors—viscoelasticity, plasticity, and fracture.

If the deformation and motion of a surface with a fixed shape can still be considered obtainable by manual operation, then physics-based simulation has huge advantages in modeling amorphous objects such as fluids, fire and cloud. It is nearly impossible for animators to generate realistic and non-repetitive responsive animations for these objects using a limited motion database but the simulated results can provide any possible behavior of the objects interacting with the environment. In [FM96], authors simulate the motion of fluids in a virtual environment. The motion of a fluid within a flow is described by a set of Navier–Stokes equations which determines the pressure and velocity of the simulated environment. Then the buoyant rigid objects can be easily included in to the scene moving along the formed fluid surface. Due to the physics-based nature, complex fluid motion effects such as refraction, reflection, and diffraction, together with rotational motion such as eddies and vorticity can be automatically generated by the simulation.

From the examples discussed above, physics-based simulation provides a fundamentally different approach for computer animation. This method is not to directly manipulate the objects and characters, but to make all movements the result of a physical simulation process. Therefore, physics-based characters and objects automatically interact in a physically precise manner without the need for additional motion data or scripts. In next section, we will describe different methods proposed to simulate different types of objects in several physics-based scenarios.

2 Physically-based modeling methods

2.1 Particle Systems

In [Ree83], the author introduces the particle system, a method for the modeling of a class of fuzzy objects. The general idea of particle system is generating particles with its own individual attributes and these particles are moved and transformed according to current dynamic attributes. In this paper, the author shows how to use particle system to generate the fire element of the Genesis Demo sequence for the movie Star Trek H: The Wrath of Khan, as well as the motion of fireworks, line drawing explosions and grass using different attributes settings for particles. The particle systems provides a solution to dynamically model fuzzy objects in detail. These objects are commonly represented by texture mapping real images at that time therefore are static and only look realistic in certain view points.

Seeing the benefit and potential of the particle system, some general tools for animating and rendering particle systems are implemented that permit both kinematic and dynamic control of particles. The author of [Sim90] continues to expand this tool sets by introducing a particle behavior language which includes position operations, velocity operations, and acceleration operations such as damping, spiraling, bouncing. The author also propose a parallel particle rendering system with which animators can simulate the dynamic phenomena such as wind, snow, water, and fire both conveniently and efficiently.

Particle systems are not only useful in the simulation of process of dynamic phenomena but also helpful in modeling surface. In [ST92], authors present a new model of elastic surfaces based on interacting particle systems. Like the computational models of fluids and solids the particles authors use have long range attraction forces and short-range repulsion forces and follow Newtonian dynamics. During the simulation, the surfaces are fitted to arbitrary collections of 3-D data points which are the position of the particles. In such way, these particle-based surfaces can be split, joined or extended without manual intervention but by controlling the physically conditions applied to the particles. The unique part of this paper is that authors provide a new perspective to view the particle system by constantly generating new particles and treating them as the forming material of objects. The author of [Ree83] also changes the lifetime of particles to infinite to generate the model of grass. These methods free animators from tedious work of modeling subtle details of complex object to make them look realistic and makes it possible for animators to control overall view of a model and leave details to computer programs.

2.2 Constrained Dynamics

Instead of controlling the movement of objects by assigning physically attribute directly to them, the authors of [BB88] present a modeling system featuring constraint-based control of rigid bodies. The models consist of individual elements which are rigid bodies acting in accordance with the rules of physics, and can thus exhibit physically realistic behavior. The authors find a set of constraint forces by solving an inverse dynamics problem which cause objects to act in accordance with the constraints designated in the environment. Because the models are built from a collection of primitive physically-based elements, this system can serve as a multi-purpose program to simulate the behavior of objects in various conditions. Later, authors also apply this method to flexible objects in [PB88]. In this paper, authors add two types of constraints: reaction constraints and augmented Lagrangian constraints. Reaction constraints allow flexible models to be pushed and pulled under the control of an animator. Augmented Lagrangian constraints create animation effects such as volume-preserving squashing and the molding of taffy-like substances. This paper enlarges the objects the constrain-based system can represent and give animators more freedom in the control of the movement of inflexible objects.

By partly using the constraints in connecting non-rigid pieces of a model and simulating nonrigid behavior based on global deformations, the authors of [WW90] provide a system which can achieve real-time performance on interacting with models that are complicated enough to be interesting subjects for animation. The method described in this paper gives insights on how to solve the problem of high computational expense of simulation and the difficulty of controlling simulated objects.

2.3 Rigid Body Dynamics

Former discussed approaches are simulating the movement of the objects under external forces and rarely considering the collisions between simulated objects,such as the authors of [TPBF87] simply give a random penalty forces to separate two objects from penetrating since the correctness of the dynamical behavior is not the top concern for the paper. Due to the lack of a model with accurate dynamical properties, animators have to use their intuition about the physical world in planning the motion of objects in the scene. Anomalies are noticeable even if it is just small abnormal movement since human eyes are so used to everyday physics. In [Hah88], authors implement a computer animation system to model general dynamic processes for arbitrary rigid bodies. The simulation of the dynamic interaction among rigid bodies takes into account various physical characteristics such as elasticity, friction, mass, and moment of inertia to produce rolling and sliding contacts. This new system introduces some concepts absent in previous dynamics simulation systems such as the collision analysis, automatically motion calculation, interaction with physically realistic environment. The variety of realistic animations generated with the system show promise in systems based on physical simulations becoming an integral part of animation of rigid bodies. Many researches follow after this physically-based rigid bodies simulation idea.

In [Bar89], authors propose a method for analytically calculating the forces between systems of rigid bodies in resting (non-colliding) contact. Compared to computationally expensive penalty methods for rigid bodies, analytical methods for rigid bodies give exact answers and produce differential equations that require far fewer time steps during simulation. Later, the author expand this method to curved surface in [Bar90]. However, these algorithms require solving linear systems of inequalities to get the simulation result which is an NP-hard quadratic programming problem and are more complex than solving systems of linear equations. The authors give an solution using a heuristic approach and point out these ideas are useful in in collision propagation problems. Finally, in [Bar94] authors present a faster, simpler and more reliable approach to calculate contact forces between solid objects. Compared to former approaches using linear programming, quadratic programming algorithms or constrained linear least-squares algorithms, the algorithm in this paper is not transformed into an optimization problem which avoids the need for sophisticated optimization software. With this direct method based on pivoting, it is feasible to compute solutions for f rigid body dynamics.

2.4 Rigid body Contact

In the process of simulating rigid body dynamics, one of the most important constraint is to avoid penetration between two rigid bodies. The previous paper [Bar94] has already provided a great solution on how to determinate the contact forces between contacting bodies. This method can be considered as a constraint-based method. In this kind of simulation, there is no forces applied directly on objects but only enforces a non-penetration constraint. The constraint-based method will require exact knowledge of the instantaneous state of contact between the objects. In a system where the constraints may change many times, determining the correct equations of motion for the objects means keeping track of these changing constraints, which is complex.

In [MC95], authors implement an impulse-based simulator in which all types of contact (colliding, rolling, sliding, and resting) are modeled through a series of collision impulses between the objects in contact. In contrast to constraint-based methods, there is no explicit constraints involved and all kinds of continuous contact are handled through a series of impulses applied to the objects. The simulator can produce physically plausible results for many problems at an interactive simulation speeds. The the impulse-based approach can calculate the contact in parallel because there is no global constraints involved and the dynamic simulation of the whole system can be decomposed into small pieces.

Later on, the authors of [GBF03] propose an algorithm to simulate such contact on non-convex rigid bodies emphasizing large scale problems with many frictional interactions. The authors represent the geometry with both a triangulated surface and a signed distance function defined on a grid, which enables the use of fast inside/outside tests. The authors also present a new shock propagation algorithm that allows for efficient use of the propagation method for treating contact. These algorithms are efficient to produce the correct physics simulation problems ranging from simple test cases to stacking problems with as many as 1000 non-convex rigid bodies with friction.

3 Applications of Physically-based modeling

3.1 Fluid Simulation

Fluid phenomena are prevalent in every day life such as ocean waves, wind, rain or tears in rain. These phenomena may seem simple and ordinary but it is hard and complex to simulate them to a realistic extent. Computational Fluid Dynamics has been a well established research area for a long time. The complex nature of fluid behavior like convection, diffusion, turbulence and surface tension still offer open research problems to be investigated. Smoothed Particle Hydrodynamics(SPH)[GM77] is an useful method in fluid simulation with some great properties. It is first invented to solve astrophysical problems in three-dimensional open space, but due to the similarity between the movement of these particles and the movement of liquids it can be modeled by the governing equations of the classical Newtonian hydrodynamics. The motion of the fluid is represented by the motion of particles and the surfaces of fluid are defined by the state of these particles. Therefore, Smoothed particle hydrodynamics is a mesh free method.

Since particle systems are proposed as a useful tool in simulating fuzzy objects, the particle-based Lagrangian approach and the grid based Eulerian approach have been used to simulate fluids in computer graphics. However, only a few techniques are available for optimizing the use of particle systems in interactive environment. In[MCG03], authors present a particle-based method which is also based on the SPH for interactive fluid simulation and rendering. By using special purpose kernels to increase stability and speed, the result can be generated at interactive speed.

3.2 Cloth Simulation

Early works on physically-based cloth animation such as [TPBF87] and [TF88] on deformable objects simulation formulate the cloth simulation problem as a time-varying partial differential equation which, after discretization, is numerically solved as an ordinary differential equation. The physics properties such as geometric state and mass distribution, outer forces are described in the parameters. In [BW98], authors describe a cloth simulation system that is much faster than previously reported simulation systems by choosing an implicit numerical integration method to solve equation. In this paper, the authors demonstrate that implicit methods for cloth overcome the performance limits inherent in explicit simulation methods.

Another progress in the cloth simulation is the solution for how to handling the collisions in cloths. Compared to rigid body collisions, collisions happening on the cloth usually turn to be more prevalent for all points may potentially collide with each other and the environment at any time. It is also difficult because very small inter penetrations can lead to cloth protruding from the wrong side which is visually disturbing and hard to correct. The common methods used in the simulation of rigid body collisions won't work on the cloth simulation. Fortunately, in [BFA02] authors present a collision handling algorithm that works with any method for simulating the internal dynamics (i.e. stretching, shearing, and bending) to efficiently yet robustly produce visually complex motion free from interference. The authors combine a fail-safe geometric collision method with a fast (non-stiff) repulsion force method that models cloth thickness as well as both static and kinetic friction. These methods provide an efficient simulation tool for complex cloth motion. Due to the few assumptions about the internal cloth dynamics, it is also less constraint to incorporate this method with advanced cloth models.

3.3 Shape matching

Like the papers discussed above many deformable models have been proposed after the pioneering work of [TPBF87]. In general, these approaches focus on an accurate material representation, on stability aspects of the dynamic simulation and on versatility in terms of advanced object characteristics that can be handled. However, there are not many solutions for simulating elastically deformable objects in interactive applications. The reasons include that existing models based are computationally expensive since they rely on complex material laws in conjunction with stable, implicit integration schemes and existing models are intended to simulate the dynamic object behavior as realistically as possible so the animators can not directly control their shapes. To overcome these restrictions, authors of [MHTG05] propose a technique which addresses these problems and contributes towards stable, interactive, and versatile deformable modeling. The main idea of this method is to replace energies by geometric constraints and forces by distances of current positions to goal positions. This approach does not require any pre-processing, is simple to compute and provides unconditionally stable dynamic simulations. The final simulating system can handle a reasonable number of deformable objects in real-time.

When it comes to simulating the deformation of large amount objects, the original shape matching method is still not efficient enough. The authors of [RJ07] present a geometrically based approach to settle the conflict between simulation speed and accuracy. This method start with applying the deformable shape matching dynamics as described in [MHTG05] to regular (cubic) lattices via a region-based convolution. This lattice shape matching with even modest filter widths is expensive and cost increases cubically with width. By applying the FastLSM algorithm proposed in this paper the lattice shape matching can be achieved efficiently at cost linear in the size of the lattice, and effectively independent of the region size. Consequently, the implemented simulator can simulate a large number of objects convincingly on a desktop machine.

3.4 Facial Animation

Facial modeling and animation not only has potential in bring cheaper solution for realistic effects in the entertainment industry but also can be beneficial in applications like lip reading and surgical planning. A static model of the human face is not sufficient to satisfy the demand in such applications. Ideally, the deformation of the face should be comply with the movement of muscles under the skin. The authors of [SNF05] build an anatomically accurate model of facial musculature, passive tissue and underlying skeletal structure using volumetric data. The developed algorithm can automatically compute control values of each muscle. Once the controls are reconstructed, the model can be subjected to interaction with external objects. However, this method relies on the motion capture markers to estimate the action of muscles. In [IKKP17], authors improve this approach by adding more flexible muscle activation model combined with a more efficient inverse physics solver which make building the simulation model for different people an almost entirely automatic process. In their muscle activation model, authors separate active and passive soft tissue layers and introducing sliding constraints to attach soft tissue to the bones. As a consequence, this model implements a variety of advanced animation effects and supports significant modifications of the face simulation model which include temporal dynamics due to inertia or external forces, realistic stretching, and offering full control of the simulation parameters.

3.5 Kelvinlets

Current digital sculpting tools are largely consists of geometric approaches. The physically based shape editors have long been sought by digital artists which can offer more natural and effective sculpting experience like doing pottery. However, existing physics-driven methods have some significant defect preventing them from being widely adopted. They are inherently slow due to the need to numerically solve equations of the underlying discrete deformation model. There are tedious setup steps such as volumetric meshing, boundary condition specification before the modeling process start. In [DGJ17], authors introduce a novel sculpting tool that offers interactive and physically plausible deformations. This approach is base on the regularization of the fundamental solutions to the equations of linear elasticity for specific brush-like forces (e.g., grab, twist, pinch) applied to a virtual infinite elastic space. Since this method only exploits closed-form analytical expressions and avoids the spatial discretization of elastic objects, it can achieve real-time feedback.

Later, in [DGJ18] authors introduce a new interactive tool for the generation of physically based dynamics with elastic deformations. This approach is based on the extension of the method described in earlier paper to dynamics. The resulting dynamic Kelvinlets lead to analytical closed-form expressions that define wave-like volumetric displacements. Consequently, this animate deformable models get rid of the geometric discretization, computationally intensive solve, or stability restriction usually required by existing physics-driven techniques.Therefore, elastic deformations can be evaluated rapidly both in space and time. The animators now can create elastic wave effects both conveniently and efficiently.

Finally, in [DGJ19] authors present an extension of the regularized Kelvinlet technique suited to non-smooth, cusp-like edits. This extended and more general collection of Kelvinlet deformations can provide fine control over both the sharpness and the spatial locality of the brush falloff. With these proposed Kelvinlet aim for various purposes, artists can blend between smooth and cusped solutions interactively, thus achieving a variety of brush profiles for sculpting or other production applications.

References

[Bar89] D. Baraff. Analytical methods for dynamic simulation of non-penetrating rigid bodies. *SIGGRAPH Comput. Graph.*, 23(3):223–232, July 1989.

[Bar90] David Baraff. Curved surfaces and coherence for non-penetrating rigid body simulation. *SIGGRAPH Comput. Graph.*, 24(4):19–28, September 1990.

[Bar94] David Baraff. Fast contact force computation for nonpenetrating rigid bodies. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '94, page 23–34, New York, NY, USA, 1994. Association for Computing Machinery.

[BB88] Ronen Barzel and Alan H. Barr. A modeling system based on dynamic constraints. *SIGGRAPH Comput. Graph.*, 22(4):179–188, June 1988.

[BFA02] Robert Bridson, Ronald Fedkiw, and John Anderson. Robust treatment of collisions, contact and friction for cloth animation. *ACM Trans. Graph.*, 21(3):594–603, July 2002.

[BW98] David Baraff and Andrew Witkin. Large steps in cloth simulation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, page 43–54, New York, NY, USA, 1998. Association for Computing Machinery.

[DGJ17] Fernando De Goes and Doug L. James. Regularized kelvinlets: Sculpting brushes based on fundamental solutions of elasticity. *ACM Trans. Graph.*, 36(4), July 2017.

[DGJ18] Fernando De Goes and Doug L. James. Dynamic kelvinlets: Secondary motions based on fundamental solutions of elastodynamics. *ACM Trans. Graph.*, 37(4), July 2018.

[DGJ19] Fernando De Goes and Doug L. James. Sharp kelvinlets: Elastic deformations with cusps and localized falloffs. In *Proceedings of the 2019 Digital Production Symposium*, DigiPro '19, New York, NY, USA, 2019. Association for Computing Machinery.

[FM96] Nick Foster and Dimitri Metaxas. Realistic animation of liquids. *Graphical Models and Image Processing*, 58(5):471–483, 1996.

[GBF03] Eran Guendelman, Robert Bridson, and Ronald Fedkiw. Nonconvex rigid bodies with stacking. *ACM Trans. Graph.*, 22(3):871–878, July 2003.

[GM77] R. A. Gingold and J. J. Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181(3):375–389, 12 1977.

[Hah88] James K. Hahn. Realistic animation of rigid bodies. *SIGGRAPH Comput. Graph.*, 22(4):299–308, June 1988.

[IKKP17] Alexandru-Eugen Ichim, Petr Kadleček, Ladislav Kavan, and Mark Pauly. Phace: Physics-based face modeling and animation. *ACM Trans. Graph.*, 36(4), July 2017.

[MC95] Brian Mirtich and John Canny. Impulse-based simulation of rigid bodies. In *Proceedings of the 1995 Symposium on Interactive 3D Graphics*, I3D '95, page 181–ff., New York, NY, USA, 1995. Association for Computing Machinery.

[MCG03] Matthias Müller, David Charypar, and Markus Gross. Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 154–159. Citeseer, 2003.

[MHTG05] Matthias Müller, Bruno Heidelberger, Matthias Teschner, and Markus Gross. Meshless deformations based on shape matching. *ACM Trans. Graph.*, 24(3):471–478, July 2005.

[PB88] John C. Platt and Alan H. Barr. Constraints methods for flexible models. *SIGGRAPH Comput. Graph.*, 22(4):279–288, June 1988.

[Ree83] William T Reeves. Particle systems—a technique for modeling a class of fuzzy objects. *ACM Transactions On Graphics (TOG)*, 2(2):91–108, 1983.

[RJ07] Alec R. Rivers and Doug L. James. Fastlsm: Fast lattice shape matching for robust real-time deformation. *ACM Trans. Graph.*, 26(3):82–es, July 2007.

[Sim90] Karl Sims. Particle animation and rendering using data parallel computation. *SIGGRAPH Comput. Graph.*, 24(4):405–413, September 1990.

[SNF05] Eftychios Sifakis, Igor Neverov, and Ronald Fedkiw. Automatic determination of facial muscle activations from sparse motion capture marker data. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, page 417–425, New York, NY, USA, 2005. Association for Computing Machinery.

[ST92] Richard Szeliski and David Tonnesen. Surface modeling with oriented particle systems. In *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 185–194, 1992.

[TF88] Demetri Terzopoulos and Kurt Fleischer. Modeling inelastic deformation: Viscolelasticity, plasticity, fracture. *SIGGRAPH Comput. Graph.*, 22(4):269–278, June 1988.

[TPBF87] Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. Elastically deformable models. *SIGGRAPH Comput. Graph.*, 21(4):205–214, August 1987.

[WW90] Andrew Witkin and William Welch. Fast animation and control of nonrigid structures. In *Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '90, page 243–252, New York, NY, USA, 1990. Association for Computing Machinery.