Physics-Based Deformable Object Simulation in Ubiquitous Computing Environments

Sunhwa Jung[1], Hongjun Jeon[2], Min-Hyung Choi[1]

[1] Department of Computer Science and Engineering, University of Colorado at Denver and Health Sciences Center, Campus Box 109, PO Box 173364, Denver, CO 80217, USA
sjung@ouray.cudenver.edu, Min-Hyung.Choi@cudenver.edu
[2] Department of Electrical and Computer Engineering, University of Colorado at Boulder, Campus Box 425, Boulder, CO 80309, USA
jeon@colorado.edu

Abstract. Ubiquity is a trend and vision of modern computer science. Planting physics-based deformable object simulation, which provides realism to animation and medical simulation, into ubiquitous computing environments will make a step forward to envisioned realism in the ubiquity of computation. The essential components of embedding physics-based deformable simulation to ubiquitous computational environments are adaptive simulation paradigm and data driven simulation techniques. In this paper we examine the feasibility of physics-based deformable object simulation in ubiquitous computing environments and present the possible applications of ubiquitous deformable object simulation.

1 Introduction

Ubiquity of computation is visionary [1], yet many researchers are pursuing it with great effort on the platform, infrastructures, and interface metaphor, utilizing high speed network, advanced computing power and light weight gadgets that can be carried in every place (held or worn). As a result, the interface between human behavior and computation becomes simpler and easier. In a few finger tips, people check email on a PDA or feature phones ubiquitously. This trend of ubiquity is spreading into every part of human activity because computers are being employed to various human activities including art, music, game, and health care.

However, more and more applications call for even higher realism in visualization and animation to accommodate the complexity of the lively collected data stream and fusion between real and virtual information. Physics-based deformable object simulation provides realistic animations and simulations mimicking natural behavior of soft objects such as cloth [2], human body [3, 4], and human organ [5]. The recent research results radically improved the realism of animations, games and medical training applications and simplified movie production procedure.

Embedding physics-based deformable object simulation to ubiquitous computing environments can potentially enhance the realism of almost every domain including entertainment, scientific and medical applications. So far, deformable object simulation has not been exploited in ubiquitous computing environments mostly because it requires heavy computational cost. In addition, current trends in ubiquitous computing often involve huge lively collected data that can not be easily processed in real-time. In this paper we examine the feasibility of physics-based deformable object simulation in ubiquitous computing environments. This paper presents the necessary features of physics-based deformable object simulation to be successfully deployed in ubiquitous computing environments and also discusses a probable example application of physics-based simulation in ubiquitous computing environments.

2 Features of Physics-based Deformable Object Simulation

Physics-based deformable object simulation is to generate natural deformation of soft objects in computer. It requires mathematical modeling of deformable object and dynamic engine for numerical solver and collision detection and resolution. 3D scene viewer is also requisite in order to view and inspect simulation results.

2.1 Modeling

Overall natural motion of deformable objects in computer graphics has been achieved by simulation the physical deformation of target objects. Mass-spring, FEM (Finite Element Methods) and BEM (Boundary Element Methods) are among the most commonly used strategies for building physics-based structures of deformable objects. For the purpose of real-time simulation, the most critical requirements are low latency and accurate physical reactions. The main focus of low latency is fast computations of stiffness matrix. The detailed underlying numerical techniques can be selected based on the goal of modeling. The modeling of physical systems often leads to partial or ordinary differential equations. The approaches proposed for deformable object simulation are mostly based on the Newton's Second Law of Motion. The displacement of an object is governed by the Lagrangian equation of motion;

$$m\frac{d^2\boldsymbol{q}}{dt^2} + k_d\frac{d\boldsymbol{q}}{dt} + \boldsymbol{g} = \boldsymbol{F} \tag{1}$$

where m is mass, $\boldsymbol{q}$ is position vector of a node, k_d is damping coefficient, $\boldsymbol{g}$ is the net internal force, and $\mathbf{F}$ is external forces. Solving this dynamics equation is a major part of the total computational cost for deformable object simulation.

To meet the demand of real time performance in dynamic simulation, mass-spring model is often chosen. Mass-spring system is one of the most common forms of physical system due to its simplicity and efficiency. This modeling provides little less accuracy, precision and stability, but it offers fast speed for interactive simulation. The basic concept of mass-spring system is discritizing the model into a set of mass points which are connected by springs and dampers. Depending on the connectivity of mass-spring system, physical property of model is defined. Mass-spring model has been studied and used in surgery simulation [5] because it is efficient to change geometric topology during simulation.

While mass-spring system approximates the object as a finite number of points and discretizes the equilibrium equation into each point, FEM models deformable objects more accurately using continuum mechanics. FEM consists of a mesh (geometry of object), and continuous equilibrium equation over each element (material property of object). FEM approach is relatively accurate, but typically much slower than discretized mass-spring model because it has to construct a large stiffness matrix and solves the system more frequently. Many research results techniques to speed it up such as modal analysis [4] and pre-computation [6].

BEM is also accurate, but it only deals with the surface of an object, thus BEM can easily model hollow-style deformable object because it needs no inner mesh unlikely FEM. BEM is relatively accurate and fast because it considers only boundary condition and it requires unique material properties instead of the complex and large number of springs which defines material property in mass-spring model and it needs no internal structure of FEM. The characteristics of the object can be changed intuitively by using the material property [7]. The accurate and fast reactions are achieved by pre-computation of stiffness matrix. However, BEM cannot model heterogeneous material property objects such as complex human tissues or large deformations.

2.2 Dynamics Engine

Dynamics engine for deformable object simulation consists of numerical linear system solver, collision handler, and constraint method. Regardless of modeling methods of deformable object, solving the equation of deformable objects model produces linear system to be solved numerically in most case. Solving the liner system requires intricate numerical methods such as Jacobi, Gauss-Seidel, and Conjugate gradient. Collision handler is the part of dynamics engine which model the interactions between objects. Collision handler detects all collisions of objects in one scene during a given time period and resolve the collisions without causing serious instability. Collision detection is time-consuming because of the complexity of the problem ($O(n^2)$, when n is the number of objects in scene). After the collision detection is processed, the collision detection result is analyzed and the state of all collision-involved elements should be corrected. But it is practically impossible to correct the collision state all at once because correcting one collision may introduce another collision in other places. Collision handling process has to iterate until all the objects are collision free. Constraint enforcement is also a key component in dynamics engine in order to enforce dynamically correct behavior of deformable objects. The boundary condition of deformable object cannot be modeled in a generic manner because a boundary condition is different from object to object and changes over a certain time period. In case of a boundary condition like fixed distance or angle for each object, proper dynamic constraints must be applied on time.

2.3 3D Scene Viewer

After deformable object simulation is performed, the simulated objects should be displayed in a rendered scene. 3D scene viewer must provide more than 30 frames in a second in order to ensure flicker-free continuous images. Deformable objects consist of thousands of polygons to be drawn regardless of its modeling scheme. Even though low resolution of model is used to physical property, a detailed geometric mesh is needed for realistic rendering. Processing large number of polygons requires heavy float point calculation especially in modern rendering algorithms. To improve the frame rate, often dedicated hardware (Graphics Process Unit) has been utilized.

3 Ubiquitous Computing Environments

Many devices and gadgets and many invisible computers are tethered with wireless and wired networks in physical environment in ubiquitous computing environments. Ubiquitous computing environments are heterogeneous and complex. The various censors transmit signals and the main server collects and processes them in order to provide quality services to the client devices. The operating systems for handheld devices are also diverse from Window CE, Palm OS, and Embedix.

As Weiser addressed the issues of hardware, network, and application in ubiquitous computing environments, the biggest challenge is combining devices together in applications[8]. Specifically the software requirements in ubiquitous computing environments are recounted by Niemelä and Latvakoski [9]: Interoperability, supporting heterogeneous network, supporting mobility, security, adaptability to the computing environment, ability of self-organization, and scalable content. These requirements have to be dealt in three different layers of software development; application layer, middleware, and system infrastructure.

4 Issues in Embedding Physics-based Simulation to Ubiquitous Computing Environment

Physics-based deformable object simulation has been successfully used in medical simulation, animation and gaming due to the recent improvement of hardware computing power even though they require high computational cost. But it is more challenging to embed them into ubiquitous computing environments due to the diversity of computing devices and live input data collected from the various devices. Therefore the key issues to successful inclusion of physics-based simulation in ubiquitous computing environments are the ability to control and compromise the computational complexity and visual realism and the ability to utilize the real-time data stream in the simulation loop to steer the simulation. These two conditions will enable complex numerical computation along with intricate visualization to be admissible to various ubiquitous computing environments with wide range of computing resources.

4.1 Adaptive Simulation

There are many types of computing machines from a light PDA to an invisible super computer. Since ubiquitous computing environment is not uniform, the performance of physics-based deformable object simulation varies. Complex simulation is impossible in light hand-held devices. Adaptive simulation paradigm will extend the possibility of simulations because it varies computational cost adaptively.

Adaptive simulation paradigm has been studied in two different aspects; spatial and temporal. Spatial adaptation changes the resolution of the deformable object model considering the required accuracy and computational cost [6, 10]. The multi-resolution models can be pre-computed deliberately before beginning simulation and the proper resolution model can be invoked during simulation. The resolution may be varied in part depending on the simulation situation. Spatial adaptation is not just for simulation, but also for rendering the simulated scenes for potential speed-up. Besides, collision detection has been also done using spatially adaptive manner considering the distance between camera and objects in scene [11].

Temporal adaptation manipulates the time step of simulation considering the stability of the simulators. When the simulators are stable enough to take a larger step, time step is getting stretched to maximum and when the simulators are getting unstable, time step is reduced. This technique reduces potentially the number of solving the governing equation. This method has been integrated into adaptive numerical solution methods using variable-step size techniques[12].

The other type of adaptation is altering numerical methods. There are several numerical methods in order to solve governing equations, and their computational costs are various. For example, numerical methods for solving ordinary differential equation are Euler, Runge Kutta, trapezoidal method, first order implicit, and second order implicit. Considering computational cost of the method and the stability of the model objects, appropriate method can be applied to in segmentation. This adaptation method has been employed successfully to cloth simulation [13]. These adaptation methods make physics-based simulation more adjustable to ubiquitous computing environments.

4.2 Data-Driven Dynamic Simulation

In ubiquitous computing environment, lively collected data from various sensors and cameras (e.g. a person’s position and movements, temperature, humidity, images etc) must be actively utilized. However, current physics-based simulation model often does consider the changes in parameters, or infusion of lively collected data to influence the simulation on the fly. Once initial parameters are set upfront to compute the future motion, there is no expedient method to directly control the behavior and trajectories of object in the middle of the simulation run even though simulation is completely different or deviated from the intended result. But ubiquitous computing environment is continuously and dynamically changing, and there are inevitable numerical errors in physical modeling because physical modeling techniques involve too much approximation and simplification to enhance the efficiency and programmability of simulation. Because of these, existing simulation model could be utterly inadequate without effectively handling live input data from ubiquitous computing environments.

In order to achieve an adjustable simulation in real-time, recently many dynamic data-driven simulation techniques have been studied. It is typically a hybrid simulation model utilizing lively captured data and pre-computed simulation data through blending and interpolation. It consists of three phases; obtainment, modification, and application. Usually the effectiveness of these methods is confined within the quality of acquired data set. In order to continuously adjust simulation according to live data set, simulation has to be data-driven and dynamic.

5 Case Study

To demonstrate that physics-based deformable object simulation can utilize input data from ubiquitous computing environments, we performed a patient specific heart simulation with our data-driven simulation model. This type of medical simulation could potentially assist healthcare professions to examine detailed conditions of a patient by blending generic simulation model and lively collected data. Embedding medical simulation into ubiquitous computing environments will provide on-site support for healthcare professions. For this purpose, a generic human heart has been simulated first, and then the behavior of a specific patient's heart model is further obtained by utilizing lively collected data stream. This two-phase approach is particularly beneficial in ubiquitous environments where the minimization the numerical computation is critical. In our study, we modulate a generic model to be the patient-specific by utilizing a series of sectional images of a specific patient's heart.

5.1 Controllable Simulation

Previously we developed an interactive method and interface techniques for controlling the behavior of a deformable object simulation using space-time constraints [14]. This method controls the behavior of deformable objects using a heuristic optimal path finding algorithm. The optimal path generator can control the motion to satisfy guided goals from lively collected data in ubiquitous environments.

One of the convenient ways to control a dynamic system is to affect the underling simulation using a set of external control parameters. We define a control vector, $\boldsymbol{u}$, which encodes all external influences. The objective of optimal control theory is to determine the control parameters that will cause a process to satisfy the physical constraints and at the same time minimize some performance criterion. We create an objective function which both measures how closely the simulation meets the key frames, also penalizes the system for using too much control forces. These two conditions can be written as:

$$\boldsymbol{J} = \frac{\boldsymbol{1}}{\boldsymbol{2}} \sum_{i=0}^{N-1} \left[\boldsymbol{\beta} \left\| \boldsymbol{q}_N^{\#} - \boldsymbol{q}_i \right\|^2 + \boldsymbol{\gamma} \left\| \boldsymbol{u}_i \right\|^2 \right] \tag{2}$$

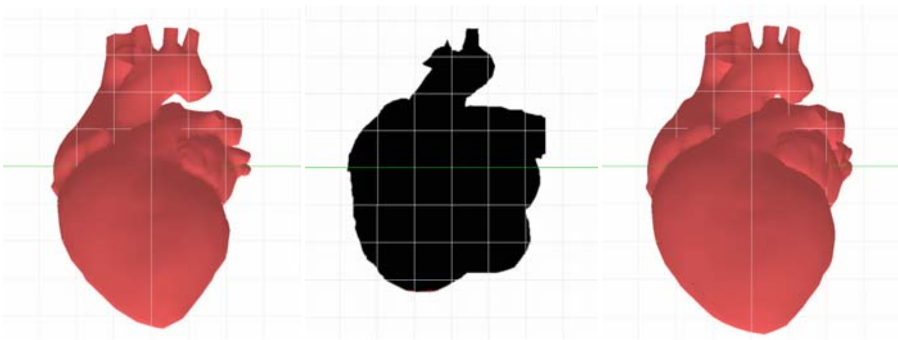
where $\boldsymbol{q}_N^{\#}$ is a desired keyframe, $\boldsymbol{q}$ is the state of a system, $\beta \geq 0$ and $\gamma \geq 0$ are weighting factors. Instead of solving the problem with the conventional nonlinear optimization method that involves expensive gradient computation, our algorithm tries to find local radial vectors only for the new control force direction that can improve the cost function every iteration step independently.

5.2 Patient-specific Heart Simulation

In our study, we assume a fast and reliable network for the lively collected image data (e.g. Computed tomography, or Magnetic Resonance Imaging) through various medical devices. Mass-spring model is used for simplicity and ease of use, but our control method can be applied to any other modeling methods. Our mass-spring model for controllable simulation is similar to our previous model used for modeling human organs [5].

We applied controllable simulation into heart simulation in order to generate patient-specific heart model. First, a detailed generic heart motion is generated off-line since it can be pre-processed from a generic geometry of a heart. Then, the sectional images of a specific patient is used as a target configuration and key frames for the specific patient's model and time-encoded motion by applying our controllable simulation scheme. Figure 1 illustrates our patient-specific simulation using a probable patient-specific model image. Our controllable simulation method modifies and transfers a generic model and its motion into guided motion of the patient's heart model.

Our pilot study on the controllable simulation and patient-specific motion transfer shows that dynamically collected data can be infused into the physics-based simulation model and well-suited the current requirements of ubiquitous computing environments. The visual display of the result can also be conveyed to properly-equipped handheld devices to further improve the ubiquity of the application.



(a) (b) (c)

Fig. 1. These figures show a patient specific heart simulation. (a): A snapshot of a generic heart motion. (b): A probable patient specific sectional image. (c): The modified motion when a part of the heart was guided by patient specific geometric changes.

6 Conclusion and Future Studies

In this paper, we examined the possibility of physics-based deformable object simulation in ubiquitous computing environments and presented a feasible example of medical simulation. Adaptive simulation scheme and data-driven simulation techniques are the key components to integrate deformable object simulation in ubiquitous computing environments.

As our future directions, we believe human factor study must be integrated in order to validate the adaptive simulation techniques because human perception plays an important role in defining the boundary resolution of models in different situations. Objective measurements for the perceived realism and the level of satisfaction also must be clearly defined to quantify and conduct comparative evaluations.

References

1. Weiser, M.: The Computer for the 21st-century. Scientific American **265** (1991) 94-104
2. Choi, K.-J., Ko, H.-S.: Stable but responsive cloth. Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques. ACM Press, San Antonio, Texas (2002)
3. Teran, J., Sifakis, E., Blemker, S.S., Ng-Thow-Hing, V., Lau, C., Fedkiw, R.: Creating and simulating skeletal muscle from the visible human data set. IEEE Transactions on Visualization and Computer Graphics **11** (2005) 317-328
4. James, D.L., Pai, D.K.: DyRT: Dynamic response textures for real time deformation simulation with graphics hardware. ACM Transactions on Graphics **21** (2002) 582-585
5. Jung, S., Hong, M., Choi, M.: Volume-preserved human organs for surgical simulation. Proceedings of the Central European Multimedia and Virtual Reality Conference (CEMVRC 05) (2005)
6. James, D.L., Fatahalian, K.: Precomputing interactive dynamic deformable scenes. ACM Transactions on Graphics **22** (2003) 879-887
7. James, D.L., Pai, D.K.: ArtDefo: accurate real time deformable objects. Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques. ACM Press/Addison-Wesley Publishing Co. (1999)
8. Weiser, M.: Some computer science issues in ubiquitous computing. Vol. 36. ACM Press (1993) 75-84
9. Niemelä, E., Latvakoski, J.: Survey of requirements and solutions for ubiquitous software. Proceedings of the 3rd International Conference on Mobile and Ubiquitous Multimedia. ACM Press, College Park, Maryland (2004)
10. Debunne, G., Desbrun, M., Barr, A., Cani, M.-P.: Interactive multiresolution animation of deformable models. Computer Animation and Simulation '99 (1999) 133-144
11. Jung, S., Hong, M., Choi, M.: An adaptive collision detection and resolution for deformable objects using spherical implicit surface. Computational Science - ICCS 2005, Pt 1, Proceedings **3514** (2005) 735-742
12. Lopez, L., Mastroserio, C., Politi, T.: Variable step-size techniques in continuous Runge-Kutta methods for isospectral dynamical systems. Proceedings of the seventh international congress on Computational and applied mathematics. Elsevier Science Publishers B. V., Leuven, Belgium (1997)
13. Boxerman, E., Ascher, U.: Decomposing cloth. Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. ACM Press, Grenoble, France (2004)
14. Jeon, H., Choi, M.-H.: Controllable simulation of deformable objects using the heuristic optimal control method. Proceedings of 12th International Conference on Geometry and Graphics (2006)