Using neuro-fuzzy method for cloth simulation

Yu-Ju Shen, Ming-Shi Wang
Department of Engineering Science,
National Cheng-Kung University, Tainan, Taiwan..

Tel: (06)275-7575 ext. 63332, Fax: (06)2766549, e-mail:shen2887@yahoo.com.tw

Abstract

A new method to simulate physical cloth behavior is presented in this paper. Our approach provides a convincing modeling for the cloth visualization. The main contribution of this work is to use neuro-fuzzy method for simulating the physical cloth behavior and the model can simply add external force to any mass points. Experimental results demonstrate the sensible and efficiency of our proposed approach.

Keywords**:** cloth simulation, neuro-fuzzy, mass-spring systems

1. Introduction

Commonly used model approaches are differential equations [6], finite element models [1] and neural networks [12].However, all existing approaches have special characteristic. Differential equations require the definition of the physical parameters of the real object, which are usually hard to determine. If measured data exist, e.g. time series data of the objects shape under influence of external forces, this leads to inverse, mostly ill-posed non-linear problems, which are still a field of mathematical research [1]. The design of finite element models is usually very time consuming, since an appropriate structure has to be defined [5]. The existing neural network based models on the other hand, need training data for learning, which is usually not available and has to be created, e.g. by use of a predefined finite-element model [12]. The learned networks are then used to replace the finite-element models and thus speed-up the real-time simulation.

Fuzzy systems and neural networks are successfully used in the area of automatic control, data analysis, and knowledge based systems[12]. Fuzzy systems can be used to derive parameters of dynamic systems, if only vague data about the system is available. Artificial recurrent neural networks can be used to simulate the dynamic of time-dependent systems. Furthermore, neural networks can be trained to simulate the behavior of real dynamic systems. Therefore, we choose a hybrid approach for simulation of cloth. Cloth is one of flexible objects considered in the computer graphics community.

The fuzzy-system is used for the derivation of the network parameters, which defines the behavior of the simulated cloth. The artificial recurrent neural network is used for the simulation process. Furthermore, it can be used to learn or adapt the parameters of the network, if measured data exist. So, the neural network and the fuzzy system can be described separately.

Mass-spring systems have been used in Computer Graphics many years. The advantage of these systems is easy implemented and faster than that the finite element methods. They have been applied to the animation of inanimate bodies such as cloth or soft material [1, 2, 3, 5].

Due to the mesh of points can only be bended along predefined horizontal , vertical, and diagonal lines between the points, the mesh is usually unable to plausible model the wrinkle which would occur in the cloth at the table's edge.

The remainder of this paper is organized as follows: related work is described in section 2. The concept of the neuro-fuzzy system is described in section 3. The experiment result and a simply conclusions are then shown in section 4 and section 5, respectively.

2. Related Work

The most simple and intuitive way of designing a mechanical simulation system is to consider the object as being discretized into a set of vertices that interact with each other through elastic forces. A time discretization process then updates numerically the position and speed of each vertex and yields the evolution of the system. By opposition to continuous systems, particle systems work on explicit discretizations of the simulated objects.

Based on this simple idea, a large of category of particle system based simulation techniques have been worked out, which mainly differ from one another by the way the forces between the particles are computed.

The simplest models, called spring-mass models, consider a triangular mesh where the vertices are masses and the edges are springs with constant rigidity and optional viscosity. These models yield very simple computations, but are not very accurate for simulating deformable surfaces, as an array of springs cannot represent exactly the elastic behavior of a plain elastic surface.Physically based model applied in the dynamic simulation of cloth apply the force calculations between points.

Terzopoulos *et al* [3] proposed a model applicable to many deformable bodies including cloth. The model of the body is deformable subject to the theories of elasticity and plasticity. Physical models of rigidity and tension are applied to the body to enforce static shape constraints, and physical properties of the body such as its mass and damping are used to simulate dynamics. Discretization of the body creates a system of linked Lagrangian ordinary differential equations. The motion and deformation resulting from these forces is calculated by integrating through time. Terzopoulos[3]have since extended the technique by adding a rigid reference component and a deformable component which are applied in the simulation of rigidity and inelastic deformations.

The Thalmanns [8] proposed extensions to Terzopoulos' employed a refinement process which represents the cloth as a B-spline surface where the control points of the surface correspond to the point masses of the original model. Where potential inaccuracies are detected, the B-spline surface is refined to better represent the discontinuity. Refinement slowed the simulation down, but did manage to produce more accurate results. The increased computation time needed for refinement was made worse by the tensor nature of B-splines, which enforced the requirement that entire bands of the surface be refined rather than just the area of the surface at the crease.

Eberhardt *et al*[5] introduced techniques to model cloth-specific properties such as hysteresis and anisotropic behavior. The system also attempts to model the effects of surface friction, air resistance, wind and moving bodies interacting with the cloth. Hutchinson *et al* [6] to define an adaptive refinement process which concentrates effort in refining the mesh, only where it is required. The technique employs a multi-level, hierarchical mesh to represent the varying level of granularity across the cloth's surface. As well as resulting in more accurate simulation, computation times are less than simulations which use a conventional uniform mesh, because the simulation can commence with a coarse mesh and refine the coarse approximation in affected regions when required. Provot [7] observed that when simulating a hanging cloth using a mass-spring model, excessive deformation often occurs at the constraint points. To solve this problem, Provot introduced constraints based on the rate of deformation. When the rate of deformation exceeds a predefined threshold, the elongation of the springs is limited. Provot later extended the system to handle collisions and self-collisions enabling the simulation of draped cloth.

3. The neuro-fuzzy model

We use neuro-fuzzy model to describe the cloth behavior. The presented network uses a problem specific structure. The structure of the neural network was designed to speed up the simulation and learning process. The structure implements the system of differential equations, which defines the mass-spring model. Assume that a surface is discretized into a set of punctual masses and spring nodes. A 2D-mesh mass-spring model is shown in Figure 1. As the figure shown, the network is structured by alternating spring and mass layers.

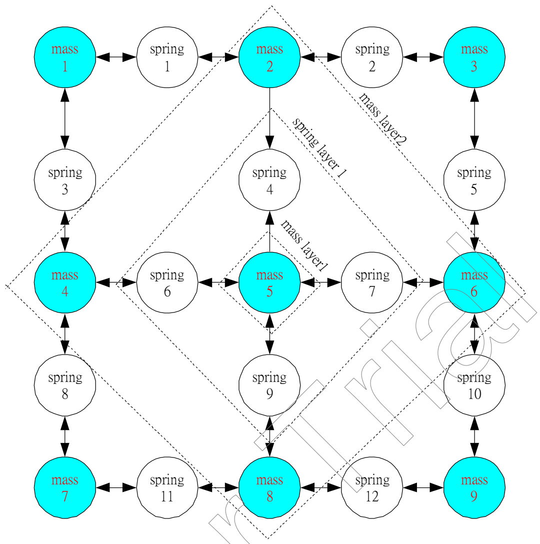

Figure 1 Network representation of a 2D-mesh

In the most common approach, the simulation loop may be divided into two steps:

Step1: Use mechanical parameters and laws to compute the vertex accelerations from the current positions and velocities of the object.

Step2: Do the integration of velocities and accelerations to compute the next vertex positions and velocities for one time step later.

Each neuron interacts by two forces, internal and external force. The internal force is used to describe the mass dynamics of the mass point, and divided in three 'sub-neurons'. These three sub-neurons are used to calculate the position, velocity, and acceleration of the mass point, respectively (see Figure 2). The applied external force is used to calculate the actual acceleration of the mass point. The velocity and position neurons are self-connected feedback nodes.

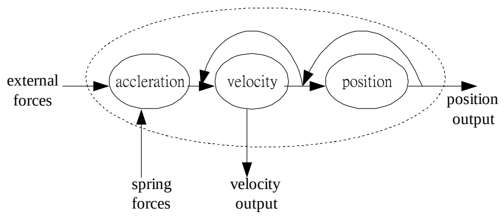

Figure 2 Description of the mass point

Let s be the number of springs and the connection matrix with *ni1* and *ni2* be the node numbers connected to spring i ,The network input, output and activation functions for the neurons are defined as[11]:

$$net_j = \sum_i \left(w_{ij} . X_i \right)$$

$$a_j = A_j(net_j) = net_j$$

acceleration neuron:

$$X_j = O_j$$

velocity and position neurons:

$$X_j = O_j \left(a_j \right) = t_c . a_j$$

Where *tc* is the time constant.

Let k be the number of the node, then the weights are initialized with the following values:

acceleration neuron:

$$w_{ij} = \begin{cases} 1 & if \quad k = n_{i1} \\ -1 & otherwise \end{cases}$$

velocity neuron:

$$w_{ij} = \frac{1}{m}$$

Where m is the mass of node.

position neuron:

$$w_{ij} = 1$$

These nodes implement the following Netowian laws, which are used for the definition of the physical spring model:

acceleration and velocity neuron

$$\sum_i F_i = F = m.a$$

velocity neuron

$$\frac{dv}{dt} = a$$

position neuron

$$\frac{dp}{dt} = v$$

The neurons behavior have been determined by two forces in the physical based models. One is the spring dynamics that can calculate the actual spring force. The other is damping force that used to simulate by the velocity vectors.

$$net_j = \sum_{i=1}^{2} \left(w_{ij} . X_i \right)$$

The activation function and the initial weights of the sub-neurons are defined as following.. The position force neuron is defined by:

$$w_{1j} = 1, w_{2j} = -1$$

The spring function

$$a_j = A_j \left(net_j \right)$$

where "*spring function"* defines the used spring function (for example, if a linear model is used a linear function which defines the spring forces). The velocity force neuron is defined by:

$$w_{1j} = 1, w_{2j} = -1$$

The damping function

$$a_j = A_j \left(net_j \right)$$

where "*viscosityfunction"* defines the used viscosity function. In case of linear models, let c be the spring constant and d be the viscosity constant, otherwise let $b_1 = b_2 = 1$. Then the force neuron is defined by:

$$w_{1j} = b_1, w_{2j} = b_2$$

$$a_j = A_j\left(net_j\right) = 1 \cdot (net_j)$$

The surface behavior can be described using several mechanical properties, which are formal models of the real behaviors of the considered material and its reactions against external interactions. The main mechanical parameters are the elasticity parameters (such as Elongation elasticity, Poisson coefficient, Bending elasticity, the viscosity, the plasticity). The parameters describing external influences mainly are the gravitation acceleration, the friction forces with the air (such as wind, turbulence and the related viscosity forces). The contact forces with colliding external objects (such as reaction forces, friction forces).

$$F = f_s(p,v) + f_d(p,v) = ma$$

where $f_s(p,v)$ defines the spring force and $f_d(p,v)$ the damping force (viscosity).

This propagation algorithm was separated in two steps.

Step1: the network was propagated until a local energy minimum was reached.

Step2: to calculate the activities a_i of the sub-neurons for the next time step.

However, the stability problems could occur during the propagation process. If the parameters of the network parameters (e.g. spring constants, force, time step, damping constants) are defined badly, the network tends to an chaotic behavior.

We are currently working on learning methods, which are based on back propagation learning methods for recurrent neural networks. The learning stage use measured data or data generated by an exact physical model for learning. The learning algorithms use the position of every node at discrete time-steps as input. In learning stage, it tries to minimize the error (total cost) function of the network. The function is defined as:

$$\begin{aligned} E &= \sum_{t=0}^{N} E(t) \\ &= \sum_{t=0}^{N}\left[\frac{1}{2}\sum_{K}\left(E_K(t)\right)^2\right] \end{aligned} \quad (18)$$

Let t = 0, 1, ..., N be the time-step, $y_k(t)$ represents the position of node k and $p_k(t)$ represents the position of the node k in the exact physical model at the time-step t, then $E_k(t)$ is defined as :

if node k have desired output $p_k(t)$ then

$$E_K(t) = p_{k(t)} - y_k(t) \quad (19)$$

otherwise $E_K(t) = 0$

The parameters (*weights*) of the network are adapted after each time-step by a gradient descent method. The learning process is finished if E is sufficient small. The fuzzy system can be used to describe the relations between existing (vague) expert knowledge of the solid behavior (for example ‘very hard’, ‘soft’, ‘elastic’) and the network parameters.

To be able to use a-priori knowledge for the in1itialization of the network parameters we use a fuzzy system. A fuzzy system approximates an unknown function based on vague samples, which are described by linguistic rules, so-called fuzzy rules[11].The fuzzy rules make use of linguistic terms defined by fuzzy sets to describe vague data. A fuzzy system consists of r parallel rules. To calculate the output of this system, the outputs of every rule are computed first. Then, all outputs are combined into a single system output. To construct a fuzzy system, the fuzzy rules as well as the membership functions describing the fuzzy sets have to be defined. The constructed fuzzy system describes the relations between the vague knowledge of the cloth (e.g. 'very hard', 'soft', 'elastic') and the network parameters. As input values (domains) we currently use mass and forces. The fuzzy system can be easily extended to different domains.

As output values the physical parameters of the network model are determined, those are spring constant, mass, viscosity and fraction force. Every domain was partitioned by fuzzy sets representing the linguistic terms used.

Some sample fuzzy rules for the description of cloth materials are as following:

- *if* material mass *is* big *and* elasticity *is* big, *then* spring constant *is* low.
- *if* material mass *is* small, .*then* spring constant *is* very low *and* viscosity *is* high.
- *if* force *is* small, .*then* spring constant *is* very low *and* viscosity *is* high

The fuzzy rules were found out by inquiring of experts and they are optimized manually. Currently we are working on neuro-fuzzy methods to optimize the derived rule base.

4. Experimental results

Several simulation results are shown in figure 4. Table 1 shows the pseudo code of the propagation algorithm. The program was run on a Pentium III-866 machine and written in C++. The parameters used in the simulation are friction=0.1, damping value=0.8, and the time step interval=0.04. In figure 5, the damping value was set to 0.05. Figure 6 shows the result for the time step=0.01.

Table 1 Pseudo code of the propagation algorithm

$t := t_0$;
$t_c := t_\Delta$;
repeat
propagate all springs at time *t*;
propagate all nodes at time *t*;
if |max $\sum f_i$ | > |max $\sum f_{i-1}$| then
tc := tc × 0.5;
else
t := t + tc;
tc := tc × 2;
end if
until $t \geq t_0 + t_\Delta$;
/* t_0 the start time
/* $\sum f_i$ sum of the force vector values
/* $\sum f_{i-1}$ sum of the former force vector values
/* t_Δ refresh rate

5. Conclusions

In this paper, we used neuro-fuzzy network system to describe cloth simulation. The weight of the network nodes can be initialized by the mass and spring parameters. It can simply add external force to any node of the model and change the object structure freely. The simulation result shows that the proposed model can obtain a better result for cloth behavior. However, the stability of propagation must be considered carefully.

REFERENCES

[1] O. Etzmub and B.Eberhardt. Collision adaptive particle systems,pages 338-453 ,2000Computer Graphics and Applications, 2000. Proceedings. The Eighth Pacific Conference on , 2000,Page(s): 338 -453

[2] D. Baraff and A. Witkin. Large steps in cloth simulation.Computer Graphics (Proc.

of SIGGRAPH ’98), pages 43–52, 1998.

[3] D. Terzopoulos, J. Platt, and A. Barr. Elastically deformable models. Computer Graphics (Proc. of SIGGRAPH ’87),pages 205–214, 1987. [2] D. Breen, D.House, and M. Wozny. Predicting the drape of woven cloth using interacting particles. Computer Graphics (Proc. of SIGGRAPH ’94), pages 365–372, 1994.

[4] M. Desbrun, P. Schr¨oder, and A. Barr. Interactive animation of structured deformable objects. Proc. of Graphics Interface ’99, 1999.

[5] B. Eberhardt, A. Weber, and W. Strasser. A fast, flexible,particle-system model for cloth draping. IEEE ComputerGraphics and Applications, 16:52–59, 1996.

[6] D.Hutchinson, Martin Preston and Terry Hewitt,Adaptive Refinement for Mass/Spring Simulations, Computer Animation and Simulation'96

[7] X. Provot. Deformation constraints in a mass-spring model to describe rigid cloth behavior. Graphics Interface, pages147–154, 1995

[8] P. Volino, M. Courchesne, and N. Thalmann. Versatile and efficient techniques for simulating cloth and other deformable objects. Computer Graphics(Proc. of SIGGRAPH ’95), pages 137–144, 1995.

[9] P. Volino, N. Thalmann, S. Jianhua, and D. Thalmann. An evolving system for simulating clothes on virtual actors.IEEE Computer Graphics and Application, 16:42–51, 1996.

[10] J. J. Hopfield and D. W. Tank. Neural computation of decision in optimization problems. Biological Cybernetics, 52:141-152, 1985.

[11] D. Nauck, F. Klawonn and R. Kruse, *Foundations of Neuro-Fuzzy Systems*, John Wiley & Sons, Inc., New York, Chichester, et al., 1997.

[12] J.-S. R. JANG,C. T. SUN and E.Mizutani, Neuro-Fuzzy and soft computing,Prentice Hall,1997.

[13] J. J. Hopfield, “Neural networks and physical systems with emergent collective computational abilities,” in *Proc. Nat. Acad. Sci. (Vol. 79)*, pp. 2554-2558, USA, 1982.

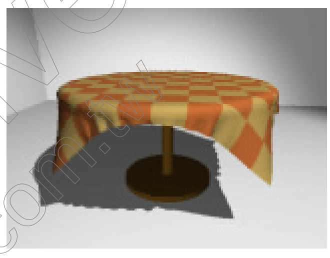

Fig.3 Snapshots from cloth simulation Friction=0.1,damping=0.03,time intervals= 0.03

Fig.4 Snapshots from cloth simulation Friction=0.1,damping=0.05,time intervals= 0.03