OPTIMAL HIERARCHICAL REPRESENTATION AND SIMULATION OF CLOTH AND DEFORMABLE OBJECTS

Konstantinos Moustakas[1,2], *Dimitrios Tzovaras*[2] *and Michael G. Strintzis*[1,2], *Fellow, IEEE*

[1]Aristotle University of Thessaloniki, 54006 Thessaloniki, Greece
[2]Informatics and Telematics Institute, P.O. Box 361, 57001 Thermi-Thessaloniki, Greece

ABSTRACT

This paper presents a novel pyramidal representation scheme for deformable object modelling, which uses a hierarchical approach to optimize the system's performance by executing the simulation in every level of a pyramid. The simulation results of each level are used to predict the lower level's state. The prediction is used as initial guess for the simulation of the lower level. The above procedure is repeated until the final level is reached. Experimental evaluation demonstrates that the proposed scheme is able to reduce significantly the computational cost, especially when the simulation involves procedures, which need large numerical computation like the Conjugate Gradient in implicit integration schemes.

1. INTRODUCTION

Deformable object modelling and simulation is a very interesting and challenging research area, while it is indispensable in applications that need very realistic simulations. Physically-based cloth animation has been under extensive study for the last decades. In the beginning of the 80's, particle systems were introduced in cloth simulation with satisfactory results [1] and offered to the researchers an alternative point of view. Since then and until '98 scientists tried to optimize and to deal with specific problems of that implementation. I.e. they tried to use alternative mass-spring systems, or to develop faster and more efficient collision detection and response algorithms.

In 1998, implicit integration was introduced [2], which was a very promising idea. In fact, implicit integration is much more stable than explicit even if the time step of the simulation is high. The trade-off for this increment of speed and stability, is the simulation accuracy. Other issues like collision detection and response have been addressed by many researchers [3] with relatively good results concerning the simulation accuracy and visual quality.

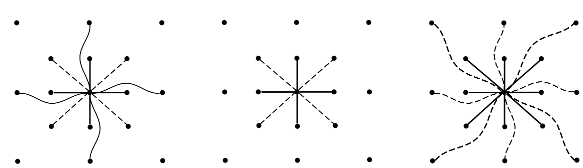
The main problem in these applications remains the computational effort needed to perform the calculations, which inhibits any attempt of real time simulation. Therefore there is still a need to emphasize in developing new models that increase the speed of the calculations. In [4], an adaptive mesh approach was introduced to emphasize in areas with high details, like folds and wrinkles. In [5], an alternative integration model was introduced, that combines the characteristics of implicit integration without the need to solve large linear systems.

Unfortunately, all the above methods decrease visual quality in order to increase the speed of the simulation. The present work aims to decrease the computational effort without a significant decrement in quality, by introducing a hierarchical representation of the object using quincunx sampling. Simulation of the mesh is performed in each level using as initial value an efficient prediction of the mesh state, generated optimally, based on the the results of the simulation in the upper level.

The paper is organized as follows. In Section 2 the mass-spring mechanical models are shortly described. The commonly used existing integration schemes are analyzed in Section 3. The novel hierarchical representation, prediction and simulation methods are discussed in Section 4. Finally Sections 5 and 6 present experimental results and conclusions, respectively.

2. PARTICLE MODEL

In the last decade the mass-spring model has been used excessively from the researchers to represent the cloth's surface. This model is assumed to be the most promising model in terms of computational complexity compared to continuous models also presented in the past. The mass-spring model consists of a lattice of mass particles linked with massless linear springs. Particles react not only to the spring forces but also to physical forces applied to them such as gravity, viscosity, damping, wind resistance, friction etc.



(a) Usual (b) Simplified (c) Alternative

Fig. 1. Mass-spring models.

In most applications, the model illustrated in Figure 1a is assumed, where each particle is linked to its 12 nearest neighbors. Structure, bending and shearing springs are the straight, dashed and curved lines, respectively. However, some variations of this approach have been also presented. The model in Figure 1b does not include bending springs. Figure 1c illustrates an alternative model, which uses eight springs (straight) to react in shrinking and eight (dashed) to react in stretching. The non-linear nature of this model may cause problems of convergence and stability. The proposed method assumes the general particle model illustrated in Figure 1a.

3. INTEGRATION SCHEMES

To animate a mass-spring system the integration schemes described in the following can be assumed.

3.1. Explicit integration

Explicit Euler integration consists of the following equations.

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^n + \frac{\mathbf{F}_i^n}{m_i} dt \tag{1}$$

$$\mathbf{x}_i^{n+1} = \mathbf{x}_i^n + \mathbf{v}_i^{n+1} dt \tag{2}$$

where $\mathbf{x}_i^n$ and $\mathbf{v}_i^n$ are the position and velocity of particle *i* in time step *n*, m_i denotes its mass, $\mathbf{F}_i^n$ is the force applied to it and dt the simulation time step.

Note that the forces at time n contribute to the velocities at time $n+1$. Thus, if the forces are relatively high or if the time step is not small enough, there will be very high and non-realistic displacements, which in the next time step will cause much higher forces. The result of this feedback is the divergence of the system. Other high order schemes, like Runge-Kutta numerical integration, can be also used to achieve more accurate and smooth results.

The advantage of these integration schemes is their simplicity and speed for each time update. But this is not enough for real-time simulations because the time step should be, in most cases, very small in order to assure convergence and stability.

3.2. Implicit integration

Implicit integration consists of the following equations.

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^n + \mathbf{F}_i^{n+1} \frac{dt}{m_i} \tag{3}$$

$$\mathbf{x}_i^{n+1} = \mathbf{x}_i^n + \mathbf{v}_i^{n+1} dt \tag{4}$$

According to the implicit integration model [3] the simulation of the mesh is described from the following equations.

$$\mathbf{v}(t+dt) = \mathbf{v}(t) + \mathbf{H}^{-1}\mathbf{Y} \tag{5}$$

$$\mathbf{x}(t+dt) = \mathbf{x}(t) + \mathbf{v}(t+dt)dt \tag{6}$$

where,

$$\mathbf{H} = \mathbf{I} - \mathbf{L}\frac{\vartheta \mathbf{F}}{\vartheta \mathbf{v}} dt - \mathbf{L}\frac{\vartheta \mathbf{F}}{\vartheta \mathbf{x}} dt^2 \tag{7}$$

$$\mathbf{Y} = \mathbf{L}\mathbf{F}(t)dt + \mathbf{L}\frac{\vartheta \mathbf{F}}{\vartheta \mathbf{x}} \mathbf{v}(t) dt^2 \tag{8}$$

and $\mathbf{L}$ is the inertia matrix, which contains in its diagonal locations the inverse masses of the particles.

In this case, the forces at time $n+1$ contribute to the velocities at time $n+1$. But as the forces at the next time step are unknown, the directional derivatives of the forces as well as their derivatives with respect to speed have to be computed, in order to generate an estimation of the forces at the next time step.

Since the predicted force at the next time step, instead of the force at the present step, is used in implicit integration, much higher time steps can be used without causing divergence.

The disadvantage of this integration scheme is the need to solve a large linear $3n \times 3n$ system (**HX=Y**), where n is the number of particles in the mesh. Fortunately, this system is sparse and efficient numerical methods like the Conjugate Gradient can be applied to reduce complexity. The computational effort needed to update the state of the system is much more higher than in explicit integration, but the overall speed is faster due to the fact that larger time steps are allowed without risking the simulation's stability.

4. HIERARCHICAL SIMULATION OF CLOTH AND DEFORMABLE OBJECTS

The computational speed provided by the modern processors is not sufficient to simulate in real time even small fabrics. Thus, a pyramidal representation of the mesh, combined with efficient prediction techniques could be very promising. This is definitely expected in cloth modelling applications. The hierarchy of the pyramid is built using quincunx sampling, due to its isotropic nature, which is assumed in order to estimate approximately the nature of the input data. The main aspects of the proposed novel algorithm will be described in the sequel.

A schematic description of the proposed simulation algorithm (for a N-level pyramid) is presented in Figure 2.

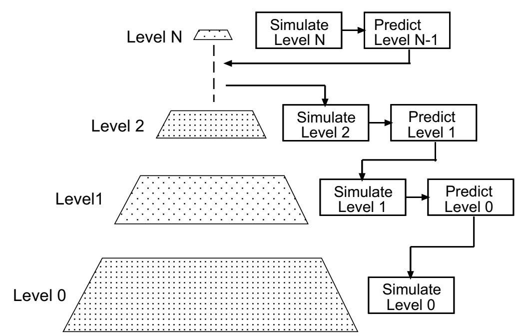


Fig. 2. Simulation pyramid

4.1. Hierarchical representation of the mesh

As already mentioned, quincunx sampling is used to build the hierarchy. Assuming $S_l[\mathbf{n}]$ to be the 2D mesh, the mesh of the upper pyramidal level is

$$S_{l+1}[\mathbf{n}] = S_l[\mathbf{M}\mathbf{n}] \tag{9}$$

where l is the level index and M the Quincunx sampling matrix

$$\mathbf{M} = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \tag{10}$$

The upsampling procedure of a mesh to a lower level lattice can be defined similarly. More precisely the upsampled output $S_{l-1}[\mathbf{n}]$ is zero unless $\mathbf{n}$ belongs to the lattice of level l.

$$S_{l-1}[\mathbf{n}] = \begin{cases} S_l[\mathbf{M}^{-1}\mathbf{n}] & , if \quad \mathbf{n} \in lattice[l] \\ 0 & , otherwise \end{cases} \tag{11}$$

There is no limit for the number of levels that can be used in the hierarchy. However, experimental results illustrate that the computational complexity does not decrease significantly if more than 5 or 6 levels are used in the hierarchy when compared to hierarchical models with a smaller number of levels.

4.2. Hierarchical implicit integration

As previously mentioned, the implementation of implicit integration schemes requires the solution of large sparse linear systems. Let us define the one-dimensional state vector of the linear system $\mathbf{H}_l\mathbf{X}_l = \mathbf{Y}_l$, based on (5-8), to be $\mathbf{X}_l = [S_l[\mathbf{n}_1], S_l[\mathbf{n}_2], ..., S_l[\mathbf{n}_N]]]$, where row scanning of the mesh lattice is assumed at level l and N denotes the total number of particles of the mass-spring model at level l.

Numerical methods like the Conjugate Gradient (CG), which are used to solve sparse linear systems, iterate until an adequate error threshold is reached. In the case of cloth simulation, if the mesh is not very small, each iteration is usually very time consuming. The idea is to use an accurate prediction of the system's solution, based on the simulation of the previous level, in order to decrease the total computational cost.

The proposed method, which utilizes the pyramidal and optimal prediction procedures, is applied as follows:

The mesh is initially simulated for the top level l. Vector $\mathbf{X}_l$ is initially set to zero and arrays $\mathbf{H}_l$ and $\mathbf{Y}_l$ are calculated utilizing the existing state of the mesh as follows.

$$\mathbf{H}_l = \mathbf{I}_l - \mathbf{L}_l \frac{\vartheta \mathbf{F}_l}{\vartheta \mathbf{v}_l} dt - \mathbf{L}_l \frac{\vartheta \mathbf{F}_l}{\vartheta \mathbf{x}_l} dt^2 \tag{12}$$

$$\mathbf{Y}_l = \mathbf{L}_l \mathbf{F}_l(t) dt + \mathbf{L}_l \frac{\vartheta \mathbf{F}_l}{\vartheta \mathbf{x}_l} \mathbf{v}_l(t) dt^2 \tag{13}$$

Next, the linear system is solved, (14), and upsampling is performed, (15), to produce the mesh of level $l-1$:

$$\mathbf{X}_l = CG\left(\mathbf{H}_l, \mathbf{Y}_l, \tilde{\mathbf{X}}_l\right) \tag{14}$$

$$\tilde{S}^0_{l-1}[\mathbf{n}] = \begin{cases} S_l[\mathbf{M}^{-1}\mathbf{n}] & , if \;\; \mathbf{n} \in lattice[l] \\ 0 & , otherwise \end{cases} \tag{15}$$

where $CG\left(\mathbf{H}_l, \mathbf{Y}_l, \tilde{\mathbf{X}}_l\right)$ is the Conjugate Gradient solver, which solves the linear system $\mathbf{H}_l\mathbf{X}_l = \mathbf{Y}_l$ using as initial guess of the solution the input vector $\tilde{\mathbf{X}}_l$ and $\tilde{S}^0_{l-1}$ is the prediction of S_{l-1} prior filtering and after upsampling.

Optimal prediction filtering follows, (16), resulting in the prediction $\tilde{S}_{l-1}$ of the mesh state at level $l-1$,

$$\tilde{S}_{l-1}[\mathbf{n}] = g[\mathbf{n}]\tilde{S}^0_{l-1}[\mathbf{n}] \tag{16}$$

The lattice $\tilde{S}_{l-1}$ is then row scanned thus producing vector $\tilde{\mathbf{X}}_{l-1}$, which is used as initial guess for the solution of the system $\mathbf{HX} = \mathbf{Y}$ at level $l-1$.

Equations (12) to (16) are executed iteratively for each level of the hierarchical representation until the base level of the mesh is reached.

4.3. Optimal prediction filtering

An issue of high importance in the presented algorithm is the prediction of the mesh state in level l of the pyramid from the data obtained from the simulation of level $l+1$. High accuracy is very important for the fast convergence of the simulation at level l. Thus a minimum variance (MV) estimation processes has to be adopted to assure adequate precision in the estimation. Besides, an isotropic spectral density model will be assumed to develop the prediction filter as will be described in the following.

The issue of prediction and filtering in pyramids constructed using Quincunx sampling has been effectively addressed in [6]. The prediction filters can be chosen in order to minimize the variance of the prediction error in each scale of the hierarchical representation. Since quincunx sampling will be used to build the pyramid, the optimum form of hierarchical prediction based on quincunx sampling is used in this paper as presented in [6, 7].

Consider the l-th hierarchy stage of a reduced pyramid Let $\mathbf{M}$ be the sampling matrix, $M = det\mathbf{M}$, $\mathbf{p}_i$ the coset vectors of $\mathbf{M}$ and $\mathbf{q}_i$ the coset vectors of $\mathbf{M}^T$, i=0,...,M-1. The coefficients are sought $g^i_l[\mathbf{Mk}]$ so as to minimize the variance of the estimation error

$$e^{(i)}_l = E\left\{\left(\tilde{S}_l[\mathbf{Mm} + \mathbf{p}_i] - S_l[\mathbf{Mm} + \mathbf{p}_i]\right)^2\right\} \tag{17}$$

where

$$\tilde{S}_l[\mathbf{Mm} + \mathbf{p}_i] = \sum_k g^{(i)}_l[\mathbf{Mk}] S_l[\mathbf{M}(\mathbf{m} - \mathbf{k})^T] \tag{18}$$

is the estimate of the ith signal polyphase component in terms of its 0-th polyphase component. These optimal filters $g^{(i)}_l[\mathbf{Mk}]$ are determined by their Fourier transform.

$$G^{(i)}_l(e^{j\mathbf{M}^T\mathbf{w}}) = \sum_{\mathbf{k}} g^{(i)}_l[\mathbf{Mk}] e^{-j\mathbf{k}^T\mathbf{M}^T\mathbf{w}} \tag{19}$$

which satisfies the equation

$$G^{(i)}_l(e^{j\mathbf{M}^T\mathbf{w}}) = \sum_{i=0}^{M-1} G_r(e^{j\mathbf{w}}) e^{-j\mathbf{p}_i^T\mathbf{w}} \tag{20}$$

where

$$G_l(e^{j\mathbf{w}}) = \frac{\Phi_l(e^{j\mathbf{w}})}{\frac{1}{M}\sum_{i=0}^{M-1}\Phi_l(e^{j(\mathbf{w}+2\pi\mathbf{M}^{-T}\mathbf{q}_i)})}$$

and $\Phi_0(e^{j\mathbf{w}})$ is the power spectral density of the pyramid input $\mathbf{x}_0$, while

$$\Phi_{l+1}\left(e^{j\mathbf{w}}\right) = \left[\frac{1}{M}\sum_{i=0}^{M-1}\Phi_l\left(e^{j(\mathbf{w}+2\pi\mathbf{M}^{-T}\mathbf{q}_i)}\right)\right]_{\mathbf{w}\to\mathbf{M}^{-T}\mathbf{w}}$$

The optimal filters can be determined alternately in the time domain by

$$R_l[\mathbf{Mt} + \mathbf{p}_i] = \sum_{\mathbf{k}} g^{(i)}_l[\mathbf{Mk}] R_l[\mathbf{M}(\mathbf{t} - \mathbf{k})] \,. \tag{21}$$

Assuming the simplified isotropic spectral density model [6],

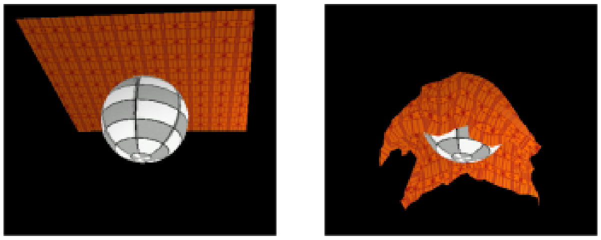
$$R_0[k_1, k_2] = A\lambda^{\sqrt{k_1^2 + k_2^2}} \tag{22}$$

an approximate solution may be obtained if a finite number of the coefficients $g^{(1)}_0[-k_1 + k_2, k_1 + k_2]$ are retained, with the rest assumed zero, and (21) is solved by least-squares techniques. If specifically only the coefficients $g^{(1)}_0[r, s]$ with $\|r\| \le 2$, $\|s\| \le 2$, are retained, and $\lambda = 0.95$ is assumed, equation (21) gives

$$\begin{aligned} G_0(z_1, z_2) = {} & 1 + 0.2992(z_1 + z_1^{-1} + z_2 + z_2^{-1}) \\ & -0.0245(z_1^{-1}z_2^2 + z_1z_2^2 + z_1^{-1}z_2^{-2} + z_1z_2^{-2}) \\ & -0.0245(z_2^{-1}z_1^2 + z_2z_1^2 + z_2^{-1}z_1^{-2} + z_2z_1^{-2}) \\ & +0.00527(z_1^3 + z_1^{-3} + z_2^3 + z_2^{-3}) \,. \end{aligned} \tag{23}$$

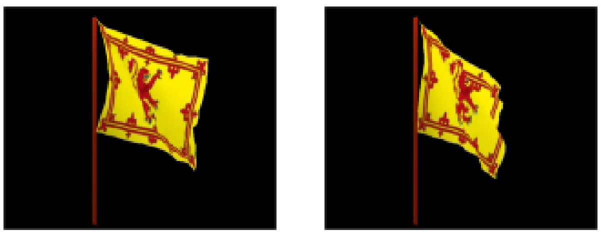
5. EXPERIMENTAL RESULTS

To illustrate the effectiveness of the proposed hierarchical simulation algorithm several tests have been performed. Figures 3 and 4 display snapshots corresponding to two of the experiments; the falling fabric and the flag simulation.



(a) (b)

Fig. 3. Falling fabric over sphere. (snapshots a) and b))



(a) (b)

Fig. 4. Flag simulation. (snapshots a) and b))

Tables 1 and 2 illustrate the computational gain of the proposed method when compared to non hierarchical schemes used in [2, 3] for the two experiments.

Computational Gain - Falling fabric (%)						
vertices	400			900		
gain	mean	min	max	mean	min	max
3 Levels	18	12	21	19	14	22
4 Levels	15	11	17	17	14	21
5 Levels	14	11	16	22	16	26

Table 1. Computational gain - Falling fabric

The hierarchical representation and simulation of deformable objects decreases significantly the computational cost. In every case, the proposed method is more efficient. It can be noticed that its efficiency is dependent from the levels used in the pyramid structure. In general if the mesh is very large, a larger pyramid structure could be more efficient than a lower. However if more levels are used in the pyramid structure than the ones needed, the computational cost may increase comparing to lower pyramids as illustrated in Tables 1 and 2 in the "400 vertices" submatrices.

It was shown experimentally that significant computational gain can be achieved using more than 2 levels of hierarchy (especially for larger meshes). Based on Tables 1 and 2 and a large number of simulations executed it can be concluded that 4 or 5 levels of hierarchy lead to optimal results.

Computational Gain - Flag simulation (%)						
vertices	400			900		
gain	mean	min	max	mean	min	max
3 Levels	21	17	27	22	17	26
4 Levels	18	15	25	21	17	24
5 Levels	17	15	22	24	21	28

Table 2. Computational gain - Flag simulation

6. CONCLUSIONS

In this paper a novel procedure for hierarchical representation and simulation of cloth and deformable objects was presented, based on pyramidal analysis. The hierarchical representation was built using quincunx sampling and optimal prediction filtering. The core of the proposed method is based on an implicit integration scheme, which uses the conjugate gradient method to solve a large sparse linear system. In each level, the mesh is simulated and after upsampling, the optimal prediction filters for quincunx sampling are used to generate an estimate of the solution of the lower level. This estimate is then used as an initial guess in order to maximize the speed of the simulation. Experimental results demonstrate that significant computational gain can be expected using the proposed method, resulting to faster implementations when compared to non hierarchical simulation schemes.

7. REFERENCES

[1] D. Terzopoulos and A. Witkin, "Physically based model with rigid and deformable components," *IEEE Computer Graphics and Applications*, pp. 41–51, Dec. 1988.

[2] D. Baraff and A. Witkin, "Large steps in cloth simlulation," in *Comput. Graph (SIGGRAPH Proc.)*, 1998, vol. 32, pp. 106–117.

[3] P. Volino and N. Magnenat-Thalmann, "Implementing fast cloth simulation with collision response," in *Computer Graphics' International*, Jul. 2000, pp. 257–266.

[4] G. Debunne, M. Desbrun, M.P. Cani, and A. H. Barr, "Dynamic real-time deformations using space & time adaptive sampling," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001, pp. 31–36.

[5] M. Meyer, G. Debunne, M. Desbrun, and A. H. Barr, "Interactive animation of cloth-like objects in virtual reality," *The Journal of Visualization and Computer Animation*, vol. 12, no. 1, pp. 1–12, 2000.

[6] D.Tzovaras and M.G.Strintzis, "Optimal construction of reduced pyramids for lossless and progreesive image coding," *IEEE Trans. on Circuits and Systems II*, vol. 47, no. 4, pp. 158–166, April 2000.

[7] I.Kompatsiaris, D.Tzovaras, and M.G.Strintzis, "Hierarchical representation and coding of surfaces using 3-d polygon meshes," *IEEE Trans. on Image Processing*, vol. 10, no. 8, pp. 1133–1151, August 2001.