10º Encontro Português de Computação Gráfica
1-3 de Outubro 2001

Wavelet Compression and Transmission of Deformable Surfaces over Networks

António Calado Lopes
antonio.calado@iscte.pt

Manuel Gamito
mag@iscte.pt

José Miguel Salles Dias
Miguel.Dias@iscte.pt

ADETTI/ISCTE, Associação para o Desenvolvimento das Telecomunicações e Técnicas de Informática, Edifício ISCTE, 1600-082 Lisboa, Portugal, www.adetti.iscte.pt

Abstract

An animation of a deformable surface with fixed topology is formed by a set of connected points (particles), which follow a trajectory over time. The transmission of such an animation is a task that consumes large amounts of bandwidth, as the position of each particle, in each time instant, needs to be transmitted.

We propose a wavelet based compression and streaming mechanism that allows the minimization of the size of the animation data, hence decreasing the total transmission time. The proposed transmission scheme, based on the time-localized feature of the wavelet transform, will be able to stream the animation so that a receiver can immediately view the fraction of the animation received, while wavelet coefficients are still arriving.

Keywords

Geometric compression, streaming, deformable models, wavelet transform

1. INTRODUCTION

In this paper, we present the specification and development of a wavelet compression, transmission and decoding scheme for deformable surfaces[1].

ADETTI has developed over the years a significant know-how in the simulation of deformable surfaces [Dias97,Dias98]. The simulations are physically based and describe the elastic behaviour of surfaces, such as cloth, that evolve over time under the action of external forces and constraints. The surfaces are represented with a triangular mesh, where the vertices of the mesh constitute a dynamic particle system, obeying physical laws. The end result of any simulation is an animation file that stores the positions of the particles for consecutive instants. We have developed a special purpose viewer, using OpenInventor components, to visualize the animation of deformable surfaces.

The next logical step, where visualization is concerned, would be to incorporate the animation data into standard VRML files and be able to access them remotely with any VRML browser. Theoretically, one could use the features of VRML that are already in place, like keyframe interpolation nodes, to represent objects whose shape changes over time. Unfortunately, the data size resulting from the physical simulation of objects is usually quite large given that the objects need a large number of particles to represent them accurately. This would lead to very large VRML files with unacceptable transmission delays through the network. We are aware that the Web3D consortium has a specific working group to discuss a possible compressed binary format for VRML that would be useful in this context [VRML]. However, no definite results have come out from that group yet. In the interim, we propose a streaming mechanism for dynamic data, much like audio or video streaming, where a remote VRML client would display the animation from the stream as it arrives, without having to wait for the end of the transmission. This specification is also applicable to other 3D geometrical formats, such as the AvatarMe .ame proprietary format, or 3D Studio MAX .3ds format [AME].

In the subsequent sections, a small introduction to the various types of deformable models will be presented, followed by an analysis of a complete compression/decompression block scheme for the type of model chosen by us. The notion of wavelet transform will be briefly analysed in order to explain how compression can be obtained. Finally, some of the compression results will be presented, as well as a future implementation of a streaming mechanism, which takes advantage of the time-localized nature of the wavelet transform in order to reconstruct an animation of a deformable model at the same time it is viewed by a receiver.

[1] This work is funded by the European Community IST Project Fashion-Me: “Fashion Shopping with individualized avatars”

2. DEFORMABLE MODELS

Since 1986, the Computer Graphics community as devoted some attention to the problem of modelling deformable objects, such as cloth or flesh, based not only in the geometrical/mathematical intrinsic characteristics of surfaces and volumes, but also in their material proprieties in the context of continuum and discrete mechanics [Feynman86, Weil86]. This approach has been included in a broader modelling framework entitled "Physically-Based Modelling", where rigid or deformable objects or even liquids and gaseous phenomena have been modelled and visualised according to the macroscopic mechanical laws that govern their static or dynamic behaviour [Barr88, Terzopoulos87, Gamito95, Stam95]. Some of the main objectives of this line of activity have been to produce images and computer animation sketches where objects show new "realistic-looking" behaviours. The inherent mathematical/mechanical modelling, include such areas as linear and non-linear dynamics or energy minimisation, whose computational complexity increase with the number of "physically-based" objects in a 3D scene and with the number of degrees of freedom that those objects show.

Interesting applications of this kind of modelling paradigm, for 3D cloth draping and synthetic fashion show simulation, have also appeared, inline with the research carried in this work. Some of the authors have developed pure geometrical methods, while others have adopted a physically-based approach, modelling cloth as a discrete set of coupled particles in interaction towards the minimisation of a potential energy, or using a Newton dynamics equilibrium formulation [Weil86, Hinds90, Feynman86, Breen94 Haumann98]. Several other researchers felt that cloth is best modelled as a continuum, where the classical macroscopic theories of elasticity or differential geometry apply and, in accordance with this idea, have developed methods for determining and visualising static or dynamic equilibrium solutions [Okabe92, Terzopoulos87, Aono90, Li95]. Mixed discrete and continuum models also exist and this has been the approach used in our modelling method: cloth can be assumed as an orthotropic linear elastic continuum, discretised by a mesh of triangles [Volino95, Dias00]. Engineers from the Materials and Textile sciences have also approached the highly non-linear problem of cloth modelling (both in geometric, as well as in elastic material terms), with the Finite Element Method achieving various degrees of success [Loyd80, Zucchini93, Eichen96]. In fact, the Textile community has developed impressive research activity in this field, starting already in 1930 that generally fall in the categories referenced before [Peirce30]. A good survey of these methods and techniques can be found in [Shanahan78, Hearle72].

Generally, we can conclude that all the authors agree in the following: physically-based simulation of deformable objects can be a CPU intensive task, and thus real-time interaction is difficult to achieve without using parallel machines and if proper modelling simplifications and numerical algorithms enhancements are not used [Baraff00, Desbrun00].

The technique reported here, enables the inclusion of deformable objects in 3D virtual environments, by exploiting the time coherence of deformable object motion and developing geometry compression/decompression and transmission methods. With our system, a user is able to interact with the scenario by means of 3D input techniques or a 3D navigation metaphor, where he can recognize, in real-time, realistic-looking (physically-based) behaviors of deformable objects. We have achieved this purpose by decoupling the modeling part of these types of objects, from the visualization/interaction parts.

In the **modeling part**, we've addressed cases of deformable surface draping over rigid body objects and we've defined linear elastic models for the computational simulation of the physically-based surfaces [Terzopoulos87]. For each surface with a known topology, we've associated a coupled particle-system and simulate its behavior by computing the dynamics of the motion of each particle belonging to the close-coupled set of the particle system. So the modeling of such objects, include also the capability to realistically visualize, in real-time, their motion evolution through time. We are able to maintain and store in a file, an history of the simulation of all the particle systems included in the 3D scene, allowing the playback of each deformable object animation at any time, at some future stage.

The simulation results file is then subjected to a coding scheme, streamed using narrow band links and sent to a remote client, where its is decoded for the **visualization/user interaction part.**

This method is quite efficient since it enables the real-time playback of the pre-computed deformable motion behavior of a number of large particle sets. With this technique, there is no need for the time consuming mechanical simulation of those sets, at the time of the deformation visualization, nor is it needed to store, at the client side, the complete history of the particle system deformation.

3. A TRANSMISSION STRATEGY

In order to compress and stream an animation of a deformable surface, a transmission strategy needs to be considered. The following figure shows the complete transmission strategy proposed.

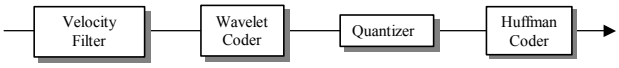

Fig. 1 – The complete encoding scheme

3.1 Pre-filtering the particle trajectories

At the beginning of the transmission scheme, we can choose to transmit any one of the time-dependent variables associated with the particles: the position, the velocity or the acceleration. In the context of a geometric visualization, only the particle positions are of interest. A deformable surface, however, can undergo large changes in its geometry, like a piece of cloth that is dropped over a table. All the position variables will exhibit, as a consequence, a very large dynamic range of values. This causes serious problems during the quantization stage of the transmission and seriously degrades the quality of the animations as seen in the browser.

The velocity is therefore a better candidate for transmission because it mostly oscillates around zero. It is a well-known fact in physics that all systems tend to evolve to minimize their internal energy. A system in such a minimum energy state will be at rest, with all the velocities being zero.

However, an even better candidate would be the acceleration given the characteristics of the systems that we have been simulating. A deformable object can be subject to very large internal stresses that also result in large forces applied at the particles. But all the forces acting at any given particle mainly cancel out each other and only a small resultant force remains. So, not only do the accelerations oscillate around zero, but they also tend to have very small magnitudes. This fact would make the acceleration the best candidate to be transmitted over the network. In what follows, we assume the time increments to be unitary. This is always true after a convenient scaling of the time variable.

The acceleration filter is described by the equation:

$$\mathbf{a}_i(k) = \mathbf{p}_i(k) - 2\mathbf{p}_i(k\text{-}1) + \mathbf{p}_i(k\text{-}2) \qquad (1)$$

where $\mathbf{a}_i(k)$ stands for the acceleration of the particle i in the time instant k and $\mathbf{p}_i(k)$ stands for the position of the particle.

However, when the acceleration filter was implemented, the results achieved weren't exactly what we expected. If we observe the signal power through the entire time span of the animation and compute the average of all trajectories powers (the variance of the signal), we can prove that low magnitude values were obtained.

$$\sigma^2 = \frac{\sum_{i=0}^{N}\left(\sum_{k=0}^{T}(a_i(k))^2\right)}{N} \qquad (2)$$

Equation (2) allows one to calculate the average power of all the accelerations vectors, given N particles and knowing that the time span of each particle ranges from 0 to T. The same formula can be applied to trajectories or velocities. Applying (2) for the acceleration vectors of a sample animation we obtain a value of $1.8\text{x}10^{-5}$.

However the acceleration originated huge oscillations around zero, thus creating spiky functions as the one shown in the next figure:

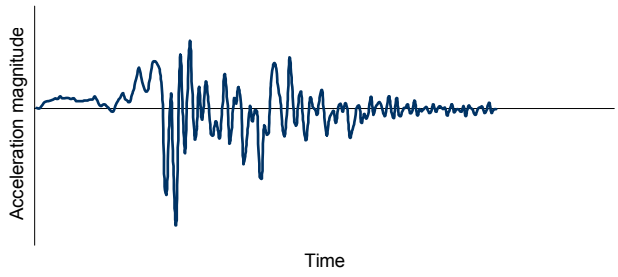

Fig.2 – The acceleration of a sample particle. This plot refers only to the particle's "x" component

This type of function is not the best to be wavelet transformed, for the cubic B-Spline wavelet[2] transform achieves excellent compression ratios with smooth functions due to the wavelet's basis function smooth nature.

Therefore we turned our attentions towards the velocity instead of the acceleration. The velocity is simply given by the first derivative of the particle's position in relation to time ($d\mathbf{p}_i/dt$), and is implemented by the equation:

$$\mathbf{v}_i(k) = \mathbf{p}_i(k) - \mathbf{p}_i(k-1) \qquad (3)$$

where $\mathbf{v}_i(k)$ stands for the velocity of particle i in the time instant k.

When transforming the trajectories in space into sets of velocity vectors we obtain a function with larger magnitude values (using (2) again we obtain $1.2\text{x}10^{-3}$ for σ^2) than the one produced by an acceleration filter, but we gain a lot in function smoothness. This fact can be easily seen in the next figure:

[2] The type of wavelet transform to be used in this work

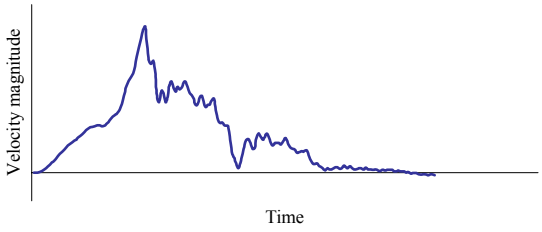

Fig.3 – The velocity of a sample particle. This plot refers only to the particle's "x" component

As our objective is to optimally compress the surface's animation, we need the smoothest function possible with the minimum amplitude range possible. The velocity is the obvious solution and the one chosen.

3.2 From velocity to trajectory

All the blocks presented in figure 1 have their own respective inverse block, which is implemented in the receiver's end. Therefore, to convert velocity vectors into trajectories one needs an inverse velocity filter that is simply implemented by the following equation:

$$\mathbf{p}_i(k) = \mathbf{p}_i(k-1) - \mathbf{v}_i(k) \qquad (4)$$

The decoding sequence must be initialized with the value $\mathbf{p}_i(0)$. The vector $\mathbf{p}_i(0)$ is the starting position of the particle, therefore, for each particle, $\mathbf{p}_i(0)$ needs to be transmitted prior to the transmission of the stream of velocity vectors. It can be included in the first packet, where the information about mesh topology is contained.

3.3 The Wavelet Transform

It is possible to further increase the transmission efficiency, by the application of a suitable transform on the stream of velocity vectors. We seek, in particular, to achieve a lossy transmission scheme, where some information is lost, in a controlled manner, without a significant loss in the end quality of the animation. Basically, this amounts to applying a linear transformation, expressed as a matrix equation that, given the sequence of velocities, returns a sequence of transform coefficients.

For a given particle i and the sequence of vectors $\mathbf{v}_i = [\mathbf{v}_i(0), \mathbf{v}_i(1),\ldots, \mathbf{v}_i(T)]$, we have:

$$\mathbf{T}\boldsymbol{\varphi}_i = \mathbf{v}_i \qquad (5)$$

where $\mathbf{T}$ is the transformation matrix and $\boldsymbol{\varphi}_i$ is the sequence of transform coefficients.

There are many transforms that can be expressed in the form (5) [Jain89]. Fourier or sine transforms could be possible options but they would require the complete set of coefficients to be transmitted before the animation could be correctly reproduced. Wavelets transforms, and the corresponding wavelet coefficients, have a much more localized nature, which enables us to generate portions of the animation on the fly, as new coefficients are being received [Strang97]. Wavelet coefficients fill a hierarchy of levels of detail. They are represented with:

$$\varphi_i^j \qquad \text{with } i = 0, \mathrm{K}\ , 2^j - 1 \qquad (6)$$

where j is the level of detail and i is the coefficient number for level j. The hierarchical structure of the coefficients becomes more evident in the following figure:

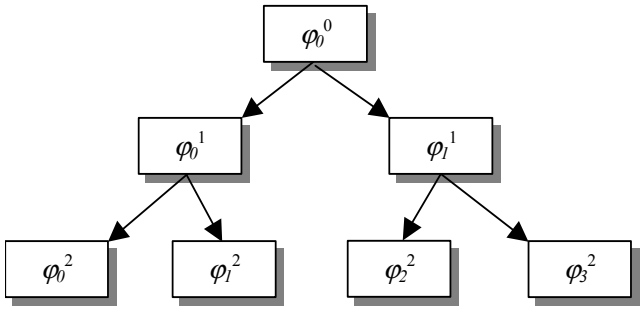

Fig. 4 - The hierarchy of wavelet coefficients.

Within the framework of wavelet transforms, there are still many wavelet bases that can be chosen to express the transformation matrix $\mathbf{T}$ in (5). One should chose a wavelet basis on the interval, in opposition to bases with infinite support, because our signals will only be valid inside some interval $[0,T]$ of time. One should also choose bases with a reasonably high number of vanishing moments. This number reflects the wavelet's ability to correctly reproduce signals. More specifically, a wavelet basis with n vanishing moments can reproduce exactly any polynomial up to degree n. The number of vanishing moments should not get too big, however, because the transform (5) becomes increasingly more expensive to compute. Good candidates for a wavelet base, according to the two previously mentioned criteria, are the n^{th} order B-spline bases on the unit interval [Stollnitz96].

The algorithm that performs the wavelet transform is depicted in the next figure:

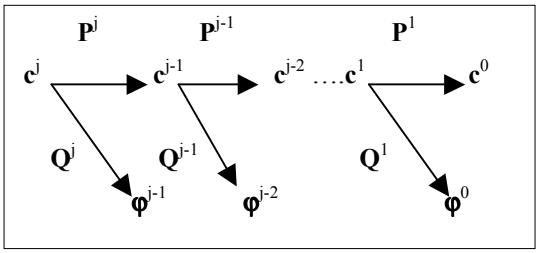

Fig.5 The filter bank

where $\mathbf{P}^j$ is the matrix containing the wavelet scaling function and $\mathbf{Q}^j$ is the matrix which represents the wavelet function itself (the mother wavelet). These two matrices, when concatenated, form the transformation matrix $\mathbf{T}$ in (5). It should be noted that, for each level of detail, there exists a unique transformation matrix, thus the set of all the transformation matrices form a filter bank. A major property of the transformation matrices is that they are mainly composed of zeroes, therefore they are sparse. This fact should be taken on account when solving (5) by using appropriate methods for sparse matrices systems, which can speed up the computation time significantly.

On the other hand, $\mathbf{c}^j$ represents the coefficients of level j, which are to be fed to the beginning of the filter bank. By the action of matrices $\mathbf{P}^j$ and $\mathbf{Q}^j$ the data is then split into a low frequency part and a high frequency part(the wavelet coefficients). Then the high frequency part is stored, and the procedure is repeated to the low frequency part, resulting again in a low and high frequency parts. The procedure is repeated until $\mathbf{c}^0$ and $\boldsymbol{\varphi}^0$ are reached. It should be mentioned that the size of the vector containing the original $\mathbf{c}^j$ must be a value belonging to the set $2^j + m$ where m is the degree of the wavelet transform (in our case, cubic B-Spline wavelets were used).

Using matrix notation, equation (5) takes the following form [Stollnitz96]:

$$\left[\mathbf{P}^j \mid \mathbf{Q}^j\right]\left[\mathbf{c}^{j-1} \mid \boldsymbol{\varphi}^{j-1}\right] = \left[\mathbf{c}^j\right]^T \qquad (7)$$

where our unknowns are $\mathbf{c}^{j-1}$ and $\boldsymbol{\varphi}^{j-1}$ and $[\mathbf{P}^j \mid \mathbf{Q}^j]$ means the concatenation of matrix $\mathbf{P}^j$ with matrix $\mathbf{Q}^j$. The concatenation should be made by interspersing the columns of both matrices in order to avoid zeros in the main diagonal of the resulting matrix. If only one zero is found in the main diagonal of matrix $\mathbf{T}$, the matrix could not be inverted leading to errors when we try to solve the system by iterative methods. By interspersing the columns of both matrices, the result is obviously mixed as well, so a reordenation is required before feeding the $\mathbf{c}^j$ to the next filter.

The method used to solve the linear system was the Bi-Conjugate Gradient Stabilized.

Wavelet compression is obtained according to a criterion of significance:

$$\left|\varphi_i^j\right| < \varepsilon \qquad (8)$$

where ε is a chosen tolerance factor. Any coefficient that obeys (8) will not be transmitted. The receiver will assume it to be equal to zero. The percentage of compression ratio is given by the number of coefficients eliminated in relation to the total number of coefficients.

Given the excellent decorrelation properties of wavelet transforms, together with their inherent locality, one can achieve significant compression ratios with very little degradation of the output signals. The result of a compression using two different tolerance factors can be seen in the next figure:

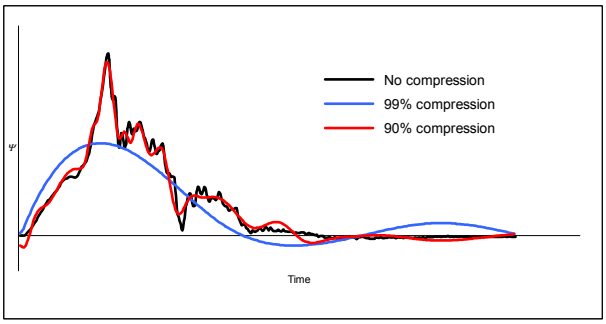

Fig.6 – A sample particle trajectory with no compression, 90% compression and 99% compression ratios.

Following the common practice in transmission schemes, the data is then submitted to a quantizer and then entropy coded using the usual Huffman encoding scheme.

3.4 The Inverse Wavelet Transform

The inverse wavelet transform allows us to recover the original signal given the wavelet coefficients. It's simply implemented by the following equation:

$$\mathbf{c}^j = \mathbf{P}^j \mathbf{c}^{j-1} + \mathbf{Q}^j \boldsymbol{\varphi}^{j-1} \qquad (9)$$

The original coefficients $\mathbf{c}^j$ are obtained by interpolating up $\mathbf{c}^{j-1}$ using the scaling function coefficients represented in the $\mathbf{P}^j$ matrix. The second term of the sum introduces a perturbation on the first term, interpolating up the wavelet coefficients [Stollnitz96]. This interpolation is obtained by the multiplication of the wavelet coefficients by $\mathbf{Q}^j$.

4. RESULTS AND DISCUSSION

By applying the complete procedure to a folding table cloth animation[3], it's possible to discern the differences between different compression ratios.

[3] A circular piece of fabric with 0.36 m of diameter (796 triangles) was placed on top of a cylinder with 0,18 m of diameter. The free surface of the fabric could bend under the gravity field. The simulation produces 400 frames, with 9.36 CPU time (secs)/frame.

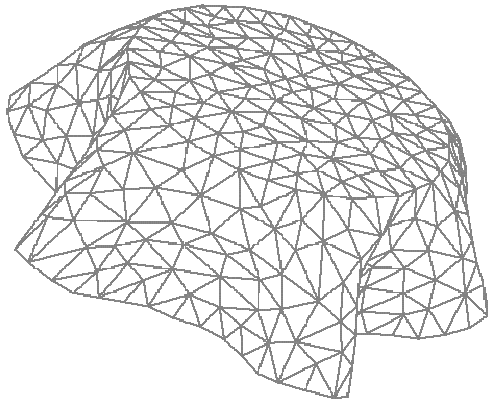

Fig.7 – A table cloth animation with no compression

In figure 7 one can see the last frame of the complete folding cloth animation with no compression. This image will be our base of comparison.

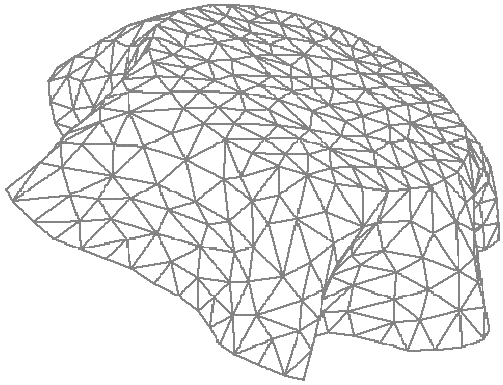

Fig. 8 – The same table cloth with 90% compression

In figure 8 it can be seen that there isn't a significant change in the geometry of the folding cloth although it was represented only with 10% of the original values, hence introducing a 90% compression ratio.

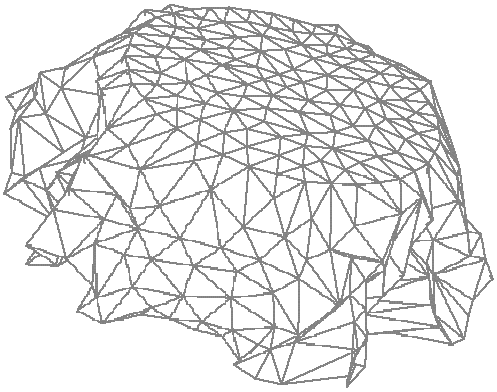

Fig.9 – The same table cloth with 99% compression

In figure 9 an exaggerated compression ratio of 99% was applied. It's evident the change in the geometry, deforming it completely, though an idea of the general animation can still be extracted.

5. FUTURE WORK: A WAVELET-BASED ADAPTIVE DISPLAY STRATEGY

In all the previous sections of this document an encoding scheme for deformable surfaces was considered. In this section it will be discussed a transmission and display strategy to be implemented in the near future.

Given the wavelet hierarchy tree depicted in figure 4, and knowing that each of the coefficients there presented refer to specific time intervals (due to the wavelet's transform time-localized nature), one can think of an effective streaming mechanism for deformable surface's animations.

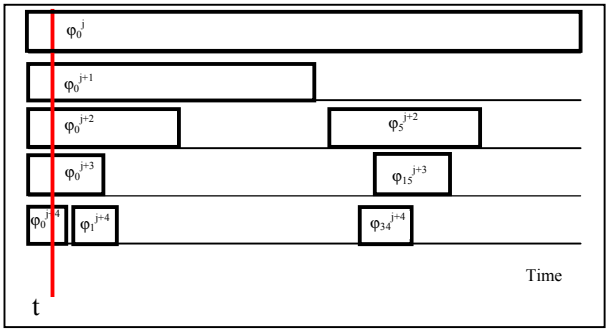

Fig 9. The wavelet coefficients of different levels involved in the representation of time instant t.

The time-localized feature of the wavelet transform is depicted in fig.9, which shows the time span associated with an arbitrary set of coefficients. Coefficients from a lower level of detail occupy a much greater length along the time axis than high level coefficients, which have a much shorter support. It is possible to see from this figure which wavelet coefficients from different levels of detail **j** are involved to represent the animation on time instant **t**. Below is the two scale relation which relates wavelets from different levels:

$$\varphi_i^j(x) := \varphi\left(2^j x - i\right) \qquad i = 0.....,2^j - 1 \qquad (10)$$

Given a specific level **j**, the mother wavelet function is scaled accordingly and translated by **i**.

From (10) we can see that coefficients at higher levels are more numerous but also have much shorter time spans. So, only a limited number of coefficients is involved in the representation of any given animation time sample.

On the other hand, the support that the mother wavelet offers follows equation (11)

$$\text{supp}\ \varphi_i^j(x) := \text{supp}\left(2^{-j}x - i\right) \quad i = 0...,2^j - 1 \quad (11)$$

or if we express (11) in interval notation:

$$\text{supp}\ \varphi_i^j(x) = \\ = \left[i2^{-j}, (m+i)2^{-j}\right] \quad i = 0..,2^j - 1 \quad (12)$$

where m is the degree of the wavelet and taking on account one of the B-Spline wavelet properties [Chui92] which states that:

$$\text{supp}\ \varphi = [0,m] \quad (13)$$

If we transmit the wavelet coefficients by traversing the wavelet hierarchy tree (Fig.4) following a depth-first, left-to-right order, we can generate portions of the animation as soon as the coefficients arrive at the receiver's end. For example, if we wanted to transmit the coefficients that support the first time instant, one should send coefficients φ_0^0, φ_0^1, φ_0^2 by this order.

This procedure can be translated as sending the low frequency components of the trajectories first (those who mostly contribute to the animation) followed by the high frequency components, which introduce detail in the animation.

The smoothness of the playback animation will be dependent on the transmission rate of coefficients. If this transmission rate is lower that the display rate of the time samples, we can choose one of two strategies:

1. Stop the animation and wait for the arrival of the next coefficients
2. Keep playing the animation with the coefficients that are available

The second strategy is possible because the animation will always be smooth no matter what the coefficients that have been received. The objects will display large scale motions, in this case, because only the large scale coefficients will be available. When the small scale coefficients finally arrive, there will be a jump in the animation as the objects are updated to their correct position.

6. CONCLUSIONS

An introduction to deformable models was presented and from all the types analysed, we have chosen a mixed discrete and continuum model applicable for plain woven cloth. Taking into consideration the problem of storing, handling and transmitting over IP networks, pre-computed deformable cloth animation, which produces large data sets, we have developed a geometry compression scheme applicable to this case. Our method requires a trajectory to velocity pre-filtering, concerning cloth particles state, to minimize the amplitude range of the original signal, corresponding to the pre-computed law of motion of the particle system. To further increase the compression rate of the pre-filtered signal, a wavelet based scheme was implemented. The cubic B-Spline wavelet was used, and we have obtained 90% compression rates without no visible degradation of the original signal.

The wavelet based adaptive strategy that was explained earlier is currently under development, therefore no results have yet been obtained. Nevertheless we are optimistic about the future performance of the transmission scheme here presented.

7. REFERENCES

[Aono90] Aono, M., "A Wrinkle Propagation Model for Cloth", CG International´90, Computer Graphics Around the World, Springer-Verlag, Tokyo, pp 95-115, 1990.

[AME] www.avatarme.com

[Baraff00] Baraff. D., Witkin, A., "Rapid Dynamic Simulation", in Cloth Modelling and Animation (edited by Donald House and David Breen), A K Peters, Ltd, pp 145-169, 2000

[Barr88] Barr, A. "Teleological Modelling", ACM SIGGRAPH Tutorial 27 Notes, 1988, B.1-B.6

[Breen94] Breen, D. E., House, D..H., Wozny, M. J., "A Particle-Based Model for Simulating the Draping Behaviour of Woven Cloth", Textile Research Journal, Vol. 64, Nr. 11, pp. 663-685, November 1994

[Chui92] Chui, C., "An Introduction to wavelets", Academic Press Limited, London, 1992

[Desbrun00] Desbrun, M., Meyer, M., Barr, A, "Interactive Animation of Cloth-Like Objects for Virtual Reality", in Cloth Modelling and Animation (edited by Donald House and David Breen), A K Peters, Ltd, pp 219-236, 2000

[Dias97] Dias, J.M.S., Galli, R., Almeida, A.C., Belo, C.A.C, Rebordão, J.M., "mWorld: A Multiuser 3D Virtual Environment", *IEEE Comp. Graphics & Appl*, 17(2), pp. 55-65, March-April, 1997.

[Dias97] Dias, J.M.S., Gamito, M.N., Rebordão, J.M., "Modelling Cloth Buckling and Drape", *Eurographics ´98 Short Presentations*, pp. 2.1.1-2.1.5, September, 1998.

[Dias00] Dias, J. M. S., Gamito, M., N., Rebordão, J. M., "A Discretized Linear Elastic Model for Cloth Buckling and Drape", Textile Research Journal 70 (4), pp 285-297, April 2000

[Eichen 96] Eichen, J, Clapp, T., "Finite-Element Modeling and Control of flexible Fabric Parts", IEEE CG&A, Vol 16-5, pp 71-80 1996.

[Feynman86] Feynman, R. *Modelling the Appearance of Cloth*, Msc Thesis, Department of Electrical Engineering and Computer Science, MIT, Cambridge, 1986

[Gamito95] Gamito, M. N., Lopes, P. F., Gomes, M. R., "Two-dimensional Simulation of Gaseous Phenomena Using Vortex Particles", In D. Thalmann e D. Terzopoulos, ed, Computer Animation and Simulation´95, pp 3-15, Springer, New York, 1995

[Haumann98] Haumann, D. R., Parent, R. E., "The Behavioral Testbed: Obtaining Complex Behaviours from Simple Rules", The Visual Computer, Vol 4, pp. 332-347 (1998).

[Hearle 72] Hearle, J, et al, "On Some General Features of a Computer-Based System for Calculation of the Mechanics of Textile Structures", Textile Research Journal, pp 613-626 (October 1972).

[Hinds90] Hinds, B. K. McCartney, J. "*Interactive Garment Design*", The Visual Computer, Vol 6-2, pp 53-61, March 1990.

[Jain89]Jain, A.K, "Fundamentals of Digital Image Processing", Prentice-Hall, Englewood Cliffs, N.J., ISBN 0133361659, 1989.

[Li95] Li, Y., et al, "Physical Modelling for Animating Cloth Motion", in *Computer Graphics: Development in Virtual Environments* (Proc. CGI) R. Earnshaw and J. Vince, eds, Academic Press, UK, pp 461-474, 1995.

[Loyd80] Loyd, D., "The Analysis of Complex Fabric Deformations", in *Mechanics of Flexible Fibre Assemblies* (edited by Hearle, J., Thwaites, J., Amirbayat, J., NATO ASI Studies), Sijthoff & Noordhoff, Alphen aan den Rijn, Holland, pp 311-342, 1980

[Okabe92] Okabe, H., et al, "Three Dimensional Apparel CAD System", Computer Graphics 26, 2 (Proc. ACM SIGGRAPH 1992), pp 105-110 (July 1992).

[Peirce30]Peirce, F. T. "The Handle of Cloth as a Measurable Quantity", J. Textile Inst., 21 pp 377-416 (1930).

[Shanahan78] Shanahan, W., J., Loyd, D. W., Hearle, J., W., S., "Characterising the Elastic behaviour of Textile Fabric in Complex Deformations", Textile Research Journal, pp 495-505 (September 1978).

[Stam95] Stam, J., Eugene, F. "Depicting Fire and Other Gaseous Phenomena Using Diffusion Processes", University of Toronto, *Computer Graphics Proc., ACM SIGGRAPH 1995*, 129 – 136

[Stollnitz96] Stollnitz, E., Derose, T. & Salesin, D. H. (1996). "Wavelets for Computer Graphics". S.Francisco:Morgan Kaufmann, ISBN 1-55860-375-1

[Terzopoulos87] Terzopoulos, D., Platt, J.,Witkin, A., Fleicher, K. "Elastic Deformable Models", Schlumberger Palo Alto Research, Computer Graphics Proc., ACM SIGGRAPH 1987, 205 – 214

[Strang97] Strang, G., Nguyen, T., "Wavelets and Filter Banks", Wellesley-Cambridge Press, Wellesley, MA., ISBN 0961408871, 1997.

[Volino95] Volino, P., Courchesne, M., Thalmann, N., "Versatile and Efficient Techniques for Simulating Cloth and Other Deformable Objects", Computer Graphics (Proc. ACM SIGGRAPH 1995), pp 99 – 104, 1995.

[VRML] The VRML Compressed Binary Format Working Group,http://www.vrml.org/WorkingGroups/vrml-bf/cbfwg.html

[Weil86] Weil, J., "The Synthesis of Cloth Objects", Computer Graphics (Proc. ACM SIGGRAPH 86), Vol. 20, N. 4, pp. 49-54, July 1986

[Zucchini93] Zucchini, R., Orsi, R., "Finite Element Analysis of Geometrically Non-Linear Shells with Snap-Through", Technical Report RT/INN/93/51, ENEA, Italy, December 1993.