Proceedings of the IASTED International Conference
PARALLEL AND DISTRIBUTED COMPUTING AND NETWORKS
February 17-19, 2004, Innsbruck, Austria

SIMULATING REALISTIC FORCE AND SHAPE OF VIRTUAL CLOTH WITH ADAPTIVE MESHES AND ITS PARALLEL IMPLEMENTATION IN OpenMP

Alam MUJAHID[1], Koh KAKUSHO[2], Michihiko MINOH[2], Yasuhiko NAKASHIMA[3],
Shin-ichiro MORI[1] and Shinji TOMITA[1]
[1] Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan.
[2] Academic Center for Computing and Media Studies, Kyoto University, Kyoto 606-8501, Japan.
[3] Graduate School of Economics, Kyoto University, Kyoto 606-8501, Japan.
Email: {mujahid, tomita}@kuis.kyoto-u.ac.jp

Abstract

Realistic simulation and manipulation of virtual cloth based on energy minimization techniques require higher computations. A model realizing force as well as shape to manipulate the virtual cloth demands even more computational cost. Mesh resolution of cloth has a great effect on the computational cost of cloth simulation. Therefore, granularity of mesh must be chosen in a manner that represents the cloth with a minimum computational cost and maintaining the quality (reality) of simulation. In this paper, we propose the adaptive mesh refinement combined with coarsening (AMRC) to have maximum possible efficiency and to maintain the acceptable reality. In addition, we propose an efficient parallel implementation in OpenMP to speed up the processing. AMRC changes the mesh density during execution time and requires the dynamic redistribution of load. A scheme of creating Active-List has been developed to manage the load balancing. We have parallelized the cloth simulation with a satisfactory efficiency by analyzing communication and synchronization overheads caused by OpenMP.

Key Words

Cloth Simulation, Adaptive Coarsening, Adaptive Refinement and OpenMP.

1. Introduction

In the community of computer graphics, visual representation of cloth has been implemented by employing the mass-spring model, finite element model and particle-based model. It has been studied that shape (position) of cloth over the time in a model is determined by using the force integration [1,2,3,4] or energy minimization [5,6,7] during animation. However, only visual information (shape) is not enough for virtual manipulation. User can only see the generated or animated images but can't feel it. Inclusion of haptic feedback (force) to represent the mechanical behavior enhances the realism. Both aspects, shape and force, are required to represent the cloth in virtual environment. It is for the first time that we have considered the relation of force with shape deformation for soft objects like cloth [6,8]. It holds the reality of shape, reality of force and reality of relation between force and shape. It utilizes the particle-based model, empirical data obtained from Kawabata Evaluation System (KES) and minimization algorithm.

Cloth has variety of visual (drape) effects and a denser mesh can only represent its realistic model that is computationally expensive. In many cases, for example containing wrinkles or interacting with user, some regions of cloth are more responsible for simulation while other regions of the mesh plays insignificant role. The efficient solution is to adaptively increase or decrease the mesh resolution in particular regions. This phenomenon introduces the concept of adaptive meshes in cloth simulation to control the computational time. Previous work [1,2,4,9,10] has used the adaptive refinement in different ways and for different applications. Later Villard [3] modified the mechanical model that is more appropriate for adaptive refinement. However, adaptive coarsening is not considered. We have already reported the comparison of adaptive coarsening and adaptive refining [11], which reflects that adaptive coarsening is better than refining. We propose in this paper, a combination of adaptive refining and coarsening. The idea is to initiate with finest mesh and then mesh density is increased / decreased adaptively during the course of simulation.

Optimization of computational time is a challenging task in the realistic simulation of deforming cloth. Therefore, speedup is an ultimate goal beside the simulation of other properties of cloth. Implementation of parallel simulator can further enhance the speed. OpenMP, a shared memory based model, helps user in implementing an easy parallel programming [12]. Communication and synchronization overheads limit the speed of a parallel program and different overheads of OpenMP are discussed in [13]. R. Lario, et al [9] has developed parallel model of multi-level cloth using OpenMP. Coarser mesh is used for overall shape representation while finer mesh is applied for small-scale features of cloth to accelerate the convergence of optimization process. This technique is similar to multi-grid model proposed in [10] and unnecessarily refines or simplifies the some region of cloth like [4], which may reduce the efficiency for very large mesh density.

It is easy for a mesh having constant density to decide the distribution of mesh among processors in advance. On the other hand, non-uniform mesh requires extra efforts to divide the mesh such that each processor has approximately same amount of work with minimum overheads. Our work uses the adaptive mesh refinement and coarsening (*AMRC*) to represent the specific features of cloth and dynamic load balancing for the parallel implementation in OpenMP to accelerate the processing.

2. Model Description

Kawabata Evaluation System (KES), a fabric-testing device, has been globally used. KES data has been utilized in [5,7] without considering the all conditions applicable during the KES measurement process. Cloth is a deformable object by stretching, bending and shearing to describe the basic property of cloth. KES characteristics describe the relation between force and shape of cloth, which are unidirectional (force to shape or vice versa) and hysteretic.

Cloth is comprised of I X J mesh of particles that employs the KES data as reference and energy minimization scheme for simulation in our model. The crossing of warp and weft thread represents a particle. We have developed cloth model by considering three main assumptions. 1) Initial state at each simulation step works as internal variable that keeps the track of previous history. 2) KES curve works as boundary value. 3) All other hysteretic cycles lie within this boundary.

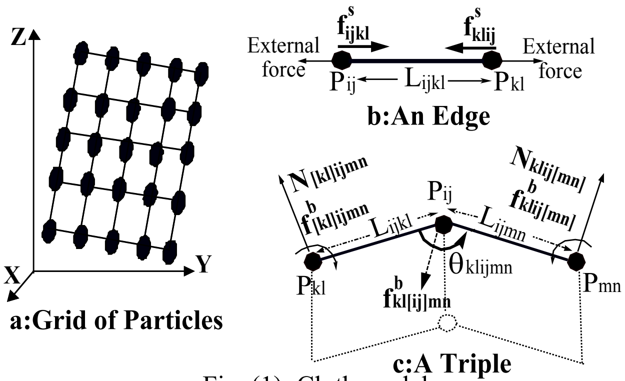

Fig. (1): Cloth model

Two adjacent particles make an edge that reflects the stretching property. Similarly three adjacent particles make a triple used to calculate the bending as shown in Fig. (1). $\mathbf{X}_{ij}$ is the position of a particle, f^s is the internal stretching force of an edge and f^b is the bending force for a triple. All other variables are function of above variables. The detail is available in [6,8].

Each element (particle, edge and triple) is characterized by energy / cost functions. We consider the three factors, motion & gravitation ($\mathbf{E}^n$), Stretching ($\mathbf{E}^s$) and Bending ($\mathbf{E}^b$), which are affecting the cloth. Since KES stretch curve is a relation from force to shape while KES bend curve describes the relation from shape to force, so stretching and bending properties are incorporated to have the relations in both directions. On the other hand Newton's law describes the bilateral relation between force and shape. Therefore, we are able to involve the force as well as shape at the same time. Other properties, for example shearing, are the same as stretch or bend and can be added in the model in the similar way. Each cost function is defined based on the KES data and represents the amount of violation from KES data. The cost is zero when the calculated data lies within or on the KES curve and cost is increasing outside the KES curve. The total cost function is the sum of individual cost functions.

$$\mathbf{E} = C_n \mathbf{E}^n + C_s \mathbf{E}^s + C_b \mathbf{E}^b$$

The values of desired variables corresponding to the equilibrium state of the cloth are obtained through the minimization process for cost functions. We have used the Polak-Ribiere Conjugate Gradient algorithm, which is faster than Steepest Descent scheme for minimization.

3. Adaptive Meshes

The trade-off between speed and realism for cloth simulation can be achieved by using the adaptive mesh size. Adaptive meshes represent the cloth region with minimum deformation (flat region) by coarser mesh and deformed region by denser mesh. Its implementation can be categorized as adaptive refinement or adaptive coarsening or combination of refinement and coarsening. We are implementing the adaptive refinement together with coarsening to make the cloth model more flexible and efficient. Implementation of adaptive meshes requires considering the following basic factors.

1. Mass conservation and force distribution.
2. Adjusting the mechanical model.
3. Updating the data structure.

Visualization of bending is more prominent than stretching in the cloth shape because stretching deforms the cloth along the line. Bending angle or curvature has been used for the mesh refinement in [2,3] and we are also using the bending cost function as criterion in this regard. When bending cost function increases from a threshold value then mesh is refined. Similarly when bending cost function is smaller, coarsening takes place. However, we are aiming to examine the cloth by refining/coarsening with respect to bending as well as stretching cost function in future. We have already adopted the adaptive refining and coarsening separately to cloth simulation and comparison is reported in [11].

3.1. Adaptive Mesh Refinement

Initial mesh density is minimum in refinement and new particles are added in some region of cloth on demand. Any new elements (Particle, Edge, Triple) must not be defined more than one time in the data structure. Refinement around a particle, adds eight new active particles and eight new ghost particles as shown in the Fig. (2). Ghost particles do not take part in simulation and are just used to maintain the topology of mesh. A new particle represents the ¼ of the area as compared to coarser particle in the above level of refinement. The length of new edge is halved in next finer mesh.

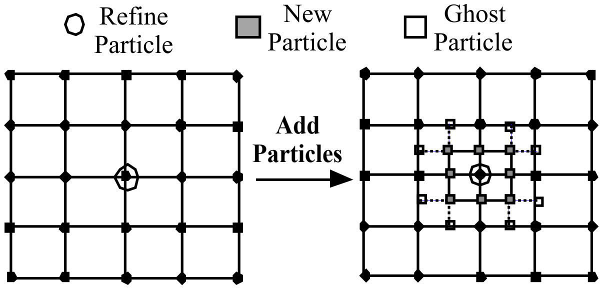

Fig (2): Adaptive mesh refinement

3.2. Adaptive Mesh Coarsening

It is the reverse operation of the refinement and initial mesh in coarsening has the maximum number of particles. It omits particles that have very small bending cost function. It does not affect the overall cost function too much. There are four edges and six triples linked with a particle, which are removed with the removal of a particle in coarsening process as shown in Fig. (3).

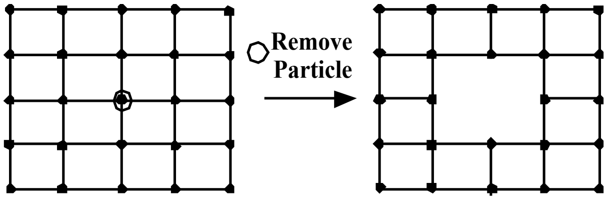

Fig (3): Adaptive mesh coarsening

3.3. Adjusting Mechanical Model

Refinement or coarsening alters the area represented by a particle. Refinement decreases the distance between two particles to half and coarsening doubles it. KES characteristic, for both bend and stretch, are available for a specific size of cloth. These characteristics must be normalized with respect to density of mesh. Therefore, our model interpolates the KES curves accordingly that makes it feasible to use KES data for adaptive meshes.

In contrast to mass-spring model, calculation of bending angle in our model involves all three particles of a triple and permits to add or omit a particle from mesh. Eliminating a particle merges a triple and an edge in weft or warp direction accordingly, for example see Fig (4). Triple ***T2*** is merged in the triples ***T1*** & ***T3***, as a result force corresponding to ***T2*** can be divided equally between ***T1*** & ***T3***. Similarly merging of edges ***E2*** & ***E3***, adds forces for resultant edge.

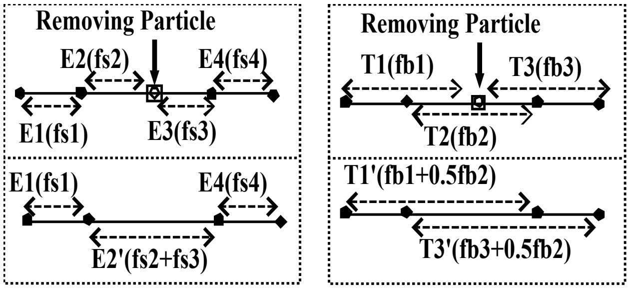

Fig (4): Force adjustment for removing a particle

As a reverse process of coarsening, refinement divides an edge into two edges having half. The space having four edges is represented by sixteen edges after refinement as shown in Fig (2). Therefore, uniform force distribution, over the area, assigns the $f^s/4$ to the finer edges. The bending parameters for refining can be adjusted in the similar manner as explained in above paragraph for coarsening but in opposite direction.

Refinement reduces the area represented by a particle to ¼ so mass should be reduced to ¼ for finer particle. In this way total mass can be preserved. However, different masses show different responses that should be tackled carefully. To have the same response, one way is to take the heavier mass as for coarser mesh and adjust it when calculating the stretching or bending. The other way is to take the lighter masses (like concept of mass density) as for finest mesh by considering that whole mesh is refined and coarser area contains more non-active / ghost particles. We are using the latter concept that also remains valid for adaptive coarsening.

3.4. Adaptive Mesh Refinement and Coarsening (AMRC)

In this work, we are implementing the adaptive refinement together with coarsening. This combination not only gives the optimum density of mesh but also makes the model more flexible and efficient. The initial cloth with flat shape is configured as maximum resolution for simulation. When it drapes over the box, deformation take places around the boundary of box. Therefore, mesh needs coarsening to lower the density in the region away from the boundary of box. Coarsening is employed provided that bending cost function is very small. In our simulation program, omitted elements (particles, edges and triples) during coarsening are not completely deleted from data structure but those are just made inactive. This information is later used for the mesh refinement when bending cost function increases than the threshold value. Therefore, in case of refining, previously inactivated particles are made active in data structure. As we are neither deleting nor adding an element in AMRC, the memory location remains unchanged. Mesh refinement and coarsening are employed accordingly after three iterations if criteria for refining/coarsening are satisfied. This process continues throughout the simulation and few possible mesh structures as an example are shown in Fig (5). The arrow symbol indicates the transition of AMRC from one stage to another.

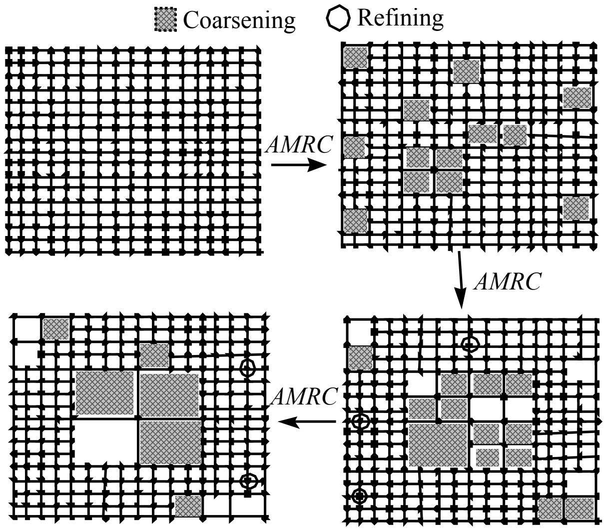

Fig (5): Adaptive mesh refinement and coarsening

4. Experimental Results

The sequential algorithm, for the rectangular cloth draping over a box, is executed on Pentium having WindowNT 4.0 operating system and OpenGL is used for displaying the results. The images with maximum (77 X 77) and minimum (20 X 20) mesh densities are shown in Fig (6) and Fig (7), which are simulated without using the adaptive meshes. Fig. (7) depicts that cloth is penetrating in the box and requires denser mesh.

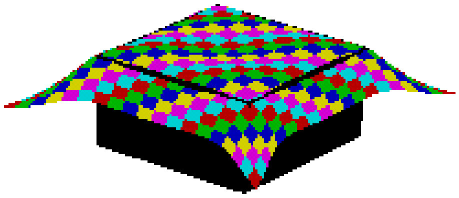

Fig (6): Simulated image with finest mesh

The cloth is simulated with adaptive refinement in next step. Mesh, having minimum initial density, is refined when bending cost function crosses the threshold value and generated image is shown in Fig (8). Similarly, the image generated by adaptive coarsening that uses the maximum density for initial mesh is shown in Fig (9). Regions having small bending cost function are coarsened.

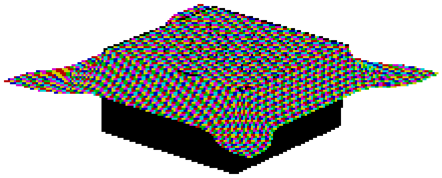

Fig (7): Simulated image with coarser mesh

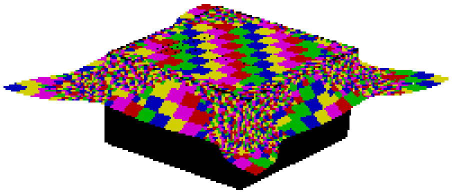

Fig (8): Simulated image with Adaptive Refinement

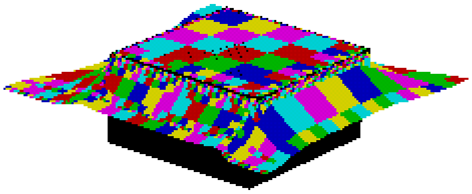

Fig (9): Simulated image with Adaptive Coarsening

Fig (10) shows the image of simulation by AMRC. Simulation process performs test whether mesh needs coarsening or refinement after three iterations. In the beginning mesh coarsening take place because most part of the cloth is flat. Later deformed elements propagate deformation to their neighbors and neighbors require refinement. Simulation is slower for early iterations due to the higher mesh density. However, computational time is satisfactory with average quality.

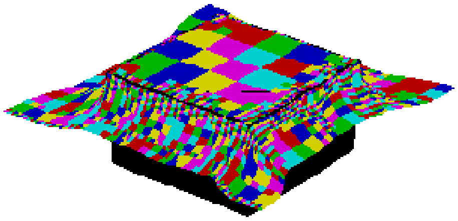

Fig (10): Simulated image with AMRC

The cost function represents the violation of cloth parameters from experimental (KES) data. The simulated cloth is seemed to be as the real one when cost function is zero. Therefore, minimized value of cost function indicates the quality of simulation. The cost value of finest mesh, used in this work, is considered as zero quality error for comparison. The comparison among different simulations in terms of processing time and quality error is given in Table 1. Table 1 depicts that AMRC produces the best quality spending the same order of processing time by the adaptive coarsening.

Table 1: Comparison for different meshes

	Time (sec)	No. Of Particles	Quality Error
Finer Mesh	2940	5939	0.0
Adaptive Refinement	1673	400 -4077	+0.057
Adaptive Coarsening	632	5939 - 1378	+0.044
AMRC	829	5939 - 1787	+0.036
Coarser Mesh	176	400	+0.071

The simulation with coarser mesh is very fast. However, quality is very poor and cloth is penetrating in the colliding object. The quality of simulation with finer mesh is very good, but it is very slow. Above two simulations indicate the extreme results and a trade-off of is achieved by applying the adaptive meshes. Adaptive coarsening is simpler as compare to refining and takes less time. Experimental results show that adaptive coarsening is better than adaptive refining in terms of computational time and quality. Adaptive refining or adaptive coarsening changes the mesh density in one direction that may cause the inaccurate representation for the complex shape of cloth. Simulation with AMRC gives the better quality and processing time is a little greater as compared with adaptive coarsening. However, acceptable increase in computational time for better quality and flexibility of model makes AMRC a good choice.

5. Parallel Implementation

Parallel system had been designed to increase the processing efficiency. A fast and easy way for parallel programming is OpenMP. We are using the OpenMP running on Fujitsu GP7000F Model 900. The implementation results show the linear speed up for the simulation of cloth.

There are three energy/cost functions relating a particle to its eight neighbors along horizontal and vertical direction in cloth structure. These tasks (cost functions) may use the same data and can't be manipulated simultaneously. Therefore, we utilize the work-sharing directive of OpenMP to keep the all tasks in one parallel region separated by event synchronization and parallelism is exploited within a task. The amount of work is assigned to a thread in such a manner that only one thread can update the shared variables in the group of nine inter-linked particles.

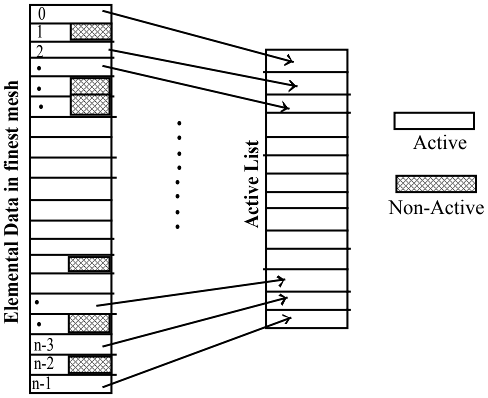

Fig (11): Active list creation

Adaptive refinement and coarsening alters the number of particles in the mesh. As mesh density is varying during execution, it requires the re-distribution of work at run time. Elements (Particles, edges and triples) are stored in data structure for finest mesh, when coarsening omits an element its status is set to be non-active. In case of refining, status of an element is changed from non-active to active. Then lists (one dimensional arrays) are created for active elements (particles or edges or triples) as procedure is shown in Fig (11). These lists contain the pointer of active elements. After coarsening procedure, non-active elements are removed from the list. Similarly active elements are inserted in the list after refining. Now the total numbers of elements in the list are distributed over the threads to balance the load. The procedure is elaborated by an example as shown in Fig (12). In this example, there are maximum 20 elements initially and only active elements are redistributed among four processors at different iterations.

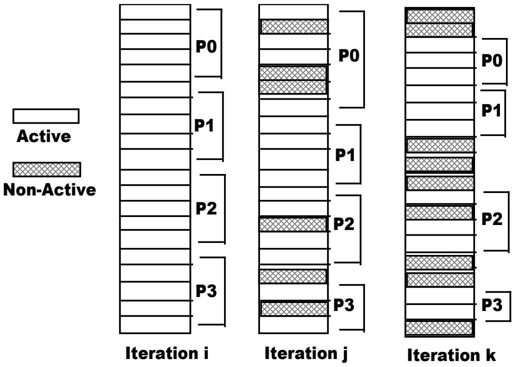

Fig (12): Work distribution among processors

Modern machines are quite complicated and parallel programming adds an extra dimension of complexity for coding. The over all performance of OpenMP depends on the coverage (percentage of code that can be made parallel), granularity, load balancing, locality and synchronization among processors. Here we are evaluating the performance in terms of computational time for the *gradient-function* (most time consuming function), *energy-function* and *total code* in the case of cloth simulated with 77 X 77 particles corresponding to different number of processors. It is clear from the Table 2 (for first 20 iterations) that parallel implementation has attained the good speedup for this simple example. We are not restricted to only 20 iterations; we have also observed the whole minimization/simulation process that gives same speedup.

Table 2: Timing analysis

No. of Threads	Gradient Time	Energy Time	Total Time	Speed Up
1	58.269698	3.878080	69.758874	1.0
2	30.175642	2.209347	37.562829	1.857
3	21.260923	1.996384	27.925706	2.498
4	16.036575	1.199117	20.745976	3.363
6	10.390235	0.816332	14.041535	4.968
8	7.901282	0.622438	11.069078	6.302

The simulation without load balancing takes 16.398753 seconds for eight processors. It means that our load balancing scheme could reduce the execution time more than 32%. Since overhead incurred by adopting load balancing is small in our situation, we expect the effectiveness of our load balancing scheme may increase with both the number of processors and size of cloth.

6. Conclusion

We have successfully implemented the adaptive mesh refinement together with adaptive mesh coarsening to simulate the force and shape of cloth. In addition to implementation of adaptive structure of mesh, we have improved the mechanical model (bend, stretch) that adopts the adaptive refining and adaptive coarsening. This simulator starts with cloth having higher mesh density that is different form other work using adaptive refinement. Therefore, simulation is slower for early iterations due to the higher mesh density. In the beginning mesh coarsening is dominant because most part of the cloth is flat. Simulation results show that we have succeeded in achieving the good computational efficiency with acceptable quality of simulation.

The combination of refining and coarsening makes our model flexible. Some applications may require refining and then coarsening or vice versa. Therefore, other applications of cloth for example, draping over irregular objects or hanging from strings, can be accommodated. Later we will extend our research work to other complex and lager size applications.

We have successfully implemented the parallel model in OpenMP and have realized the effective load balancing. Comparison of sequential processing time with parallel processing time reflects the good speedup achieved by our parallel model. Of course, this linear speedup is realized for a small size of cloth. Applying this algorithm to lager size applications (for example, human body wearing the shirt represented by 4000 X 4000 particles), we have to implement more elaborate algorithm for load balancing by considering other factors such as cache coherency. This is a problem for future work.

Currently visualization part is a negligible task in comparison with simulation part because simulation is computationally expensive and non-real time simulation has been performed at this stage. On the other hand, we also have a research project on parallel visualization of simulation results. So, we could utilize the results of this work if we could achieve the real time simulation of cloth with force feedback.

References

[1] P. Howlett and W.T. Hewitt, Mass-spring simulation using adaptive non-active points, *Computer Graphics Forum, Vol. 17*(3), 1998.

[2] D. Hutchinson, M. Preston and W.T. Hewitt, Adaptive refinement for mass/ spring simulations, *7th Eurographics Workshop on Animation and Simulation*, 1996, 31-45.

[3] Jullian Villard and Houman Borouchaki, Adaptive meshing for cloth animation, *11th International Meshing Roundtable*, 2002, 243-252.

[4] D. Zhang and M.M.F Yuen, Cloth Simulation Using multilevel meshes, *Computer and Graphics, Vol. 25*(3), 2001, 383-389.

[5] D.E Breen, D.H. House, M.J. Wozny, Predicting the drape of woven cloth using interacting particles, *Computer Graphics (Proc. SIGGRAPH)*, 1994, 365-372.

[6] Mujahid et al., A new approach towards modeling the virtual cloth based on realistic force and shape, *8th International Conference on Virtual Systems and MultiMedia,* Gyeongju, Korea, 2002, 399-408.

[7] Y. Sakaguchi et al., Party: A numerical calculation method for a dynamically deformable model, *System and Computers in Japan*, *Vol. 26*(8), 1995, 75-87.

[8] Mujahid et al., Modeling virtual cloth to display realistic shape and force based on physical data, *Journal of System, Control and Information, Japan, Vol. 47*(4), 2003, 183-191.

[9] R. Lario et al, Rapid parallelization of a multilevel cloth simulator using OpenMP, *3rd European Workshop on OpenMP*, Barcelona, Spain, 2001.

[10] H.N. Ng, R.I. Grimsdale and W.G. Allen, A system for modeling and visualization of cloth material, *Computer and Graphics, Vol. 19*(3), 1995, 423-430.

[11] Mujahid et al., Comparison between adaptive refinement and adaptive coarsening for simulating realistic force and shape of virtual cloth, *MMU International Symposium on Information and communications Technologies,* Malaysia, 2003, 147-150.

[12] R. Chandra et al, *Parallel programming in OpenMP* (Morgan Kaufmann Publishers, 2001).

[13] J.M. Bull: Measuring synchronization and scheduling overheads in OpenMP, *1st European Workshop on OpenMP*, Lund, Sweden, 1999.