

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

# INFORMATION TECHNOLOGY JOURNAL

**ANSI***net*

Asian Network for Scientific Information  
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

## Area Based Adaptive Subdivision of a Triangular Mesh for Visio Haptic Deformation

I. Ahmad and S. Sulaiman

Department of Computer and Information Science, Universiti Teknologi PETRONAS,  
Bandar Seri Iskandar, 31750, Tronoh, Perak, Malaysia

---

**Abstract:** Soft tissue simulation is very important in medical simulation and learning procedures. But such simulations require intensive computing. Besides the fact that haptic devices allow touch and feel the virtual objects which increases realism in virtual environment, this imposes much higher requirement on speed and accuracy of computation. The reason is refresh rate for realistic visualization is 25 Hz while the refresh rate of the realistic haptic rendering is 1000 Hz. In this study, an efficient approach for refining a varied area triangular mesh for Visio haptic deformation is presented. The mesh refines adaptively upon contacted by a force feedback device. The proposed algorithm guarantees smooth local deformation as the triangles with larger area are divided more as compared to small area triangles. After subdividing the contacted region, new masses, mass spring constants and dampers are assigned to maintain the original mesh properties. The proposed method is most suitable for manually designed model in 3D modeling software. We integrate our algorithm with the physics based Mass Spring Model for real time deformation.

**Key words:** Mass spring system, adaptive subdivision, deformation, real time

---

### INTRODUCTION

Real time interaction with deformable objects is one of the interesting and challenging areas of study especially with applications to medical simulation and interactive training simulators. Augmenting such simulators with haptics makes it more demanding as it requires more computation. Refresh rate of visual channel is 30 Hz while haptic channel refresh rate is close to 1 kHz (Peterlik *et al.*, 2010). Therefore, efficient algorithms are needed for interacting with soft bodies using force feedback devices.

There are two methods used in computer graphics for modeling deformable objects:

- **Geometry based:** In geometry based deformation, the geometry of the model is directly manipulated but the internal physics and dynamic interactions are not considered. In this category two approaches used are vertex based and spline based deformation
- **Physics based:** These models simulate the physical behavior of objects and consider the internal and external forces. In physics based, FEM (Finite Element Method) and Mass Spring System

are used in computer graphics for deformable objects simulation

Many techniques for surface representation are used for soft body simulation. One popular technique, among many is subdivision which achieved much attention in the recent years. Two main categories in subdivision are interpolatory and approximating schemes. In interpolatory subdivision schemes the original vertices are carried over to the next level of subdivision while in approximating schemes, the original vertices are not retained at newer levels of subdivision. Butterfly (Dyn *et al.*, 1990), Modified Butterfly (Zorin *et al.*, 1996) and Kobbelt (1996) are interpolatory subdivision schemes while Doo and Sabin (1978), Catmull and Clark (1978) and Loop are approximating. These are the popular techniques in the area of surface subdivision.

These schemes provide global refinement at every subdivision level. Practically, it is seen that some areas in a mesh become smooth after few subdivisions while other still need further subdivision to achieve smoothness. In such application, global refinement leads to a heavy computational load because of subdividing regions of a mesh which are already smooth. Adaptive subdivision removes this extra computation while maintaining the smoothness.

Adaptive subdivision first proposed by Muller and Jaeschke (1998) for Catmull-Clark and Doo-Sabin subdivision schemes. Xu and Kondo (1999) developed an adaptive scheme based on the Doo-Sabin scheme. Kobbelt (2000) revised his Kobbelt scheme by introducing adaptive subdivision and also for v3 subdivision scheme.

Adaptive refinement is also integrated with the Physics based Mass Spring System for deformable objects. An adaptive mass spring model was used for a piece of cloth which refine the regions of high curvature (Hutchinson *et al.*, 1996). O'Brien and Hodgins (1999) simulated the fracture of glass by adaptively refining the mesh at glass fractures. Kawai (2001) builds hierarchical mass spring model to achieve adaptive refinement during object deformation. Another method refines adaptively triangular mesh which is independent of regularity of the initial mesh (Choi *et al.*, 2002). The algorithm proposed by Kahler *et al.* (2003), dynamically refines the mesh at high curvature which is integrated with the mass spring model.

An adaptive subdivision is also integrated with mass spring model (Zhang *et al.*, 2002). The model refines in the region where the external force is applied by force feedback device. They used Loop subdivision scheme in the area of interest but the mesh they used in their experiment is regular.

Mass Spring Systems have been largely used for deformable objects because it is easy to implement and because of low computational complexity. However, even with Mass Spring System increasing number of nodes and springs makes the deformation difficult. But for higher resolution and realistic view more nodes are necessary. Therefore, to cope with such problem, a local refinement is a good choice. Previous methods apply local refinement to the whole selected area irrespective of the refinement needs in some of the regions. In this study, we propose an Area based adaptive subdivision algorithm based on Loop subdivision scheme which is integrated in the Physics Based deformation using Mass Spring Model. The algorithm refines the area adaptively for smooth deformation upon contacted by the force feedback device. During subdivision process, the physical properties of the deformable object are maintained. Our algorithm removes unnecessary nodes which ease extra computation in terms of spring forces and also improve smooth deformation.

**MASS SPRING SYSTEM**

In deformable body simulation, the physics behind must be quick enough to achieve real time performance.

Mass Spring Model is well suited for such simulation and has been used extensively in medical and virtual reality simulators. A mass spring model consists of point masses also known as nodes and springs. These nodes are connected to each other by springs which are source for propagating energy among these nodes. In our model, the surface is made up of triangles. Each edge of a triangle acts as a spring and each vertex as node. The spring connecting two nodes holds it in place. For our model, two meshes are maintained (Zhang *et al.*, 2002). One mesh called Fixed Mesh is for maintaining the shape of the object and the other called Deformable Mesh is used in simulating deformation. Nodes in two meshes are also connected with springs whose spring stiffness is denoted by  $k_i$  and there is no spring force on the point masses in Home Mesh. Home springs come into play when the mesh is deformed using haptic force feedback by applying force on nodes to bring it back to the original resting position. The mass spring model used is shown in Fig. 1.

The dynamics of nodes in a Mass Spring Model is governed by Newton's law of motion and is given by:

$$m_i \ddot{u}_i + c_i \dot{u}_i + k_i u_i = F_{int} + F_{ext} \tag{1}$$

where,  $u_i$ ,  $\dot{u}_i$  and  $\ddot{u}_i$  are the position, velocity and acceleration vectors of node  $i$ , respectively.  $m_i$ ,  $c_i$  and  $k_i$  are scalar and represent mass, the damping and stiffness coefficient of node  $i$ .  $F_{int}$  is the spring force vector acting on node  $i$ , due to springs connecting node  $i$  and  $F_{ext}$  is the external force vector exerted on node  $i$ , through a haptic force feedback device.

Rest length  $l$  of all the springs in the model that define the original shape of the model, is pre computed and stored. When a node change its position from  $u_1$  to  $u_2$  and its velocity from  $v_1$  to  $v_2$  then the internal force  $F_{int}$  can be calculated as:

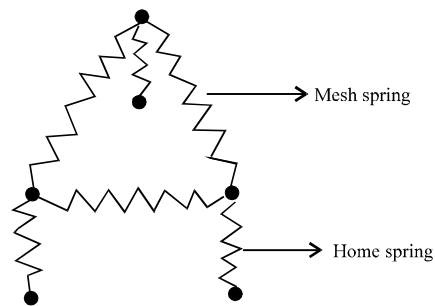


Fig. 1: Mass spring model with home and mesh spring

$$F_{int}^1 = - \left[ k_s (|\Delta u| - l) + k_d \left( \frac{\Delta v \cdot \Delta u}{|\Delta u|} \right) \right] \frac{\Delta u}{|\Delta u|} \quad (2)$$

$$F_{int}^2 = -F_{int}^1$$

where,  $F_{int}^1$  and  $F_{int}^2$  are the spring forces on node 1 and 2, respectively. In Eq. 2,  $k_s$  is a spring constant and  $k_d$  is a damping constant.  $l$  is the rest length of the spring.  $\Delta u = u_1 - u_2$  is the actual distance between two nodes and  $\Delta v = v_1 - v_2$  is the difference of velocities of two nodes. Spring force is proportional to the difference of length of the spring and damping force is proportional to the difference of velocity of the node.

There are many techniques which are used for numerical solution. For the numerical solution of Eq. 2, implicit or explicit methods may be used. If the time step is large then implicit methods are good candidates. In deformable and soft object simulation, time step must be small enough so that smooth deformation is achieved. In this case explicit methods are good which guarantees smooth deformation. Here we used Runge-Kutta order 4 methods. General form of Runge-Kutta equation is given as:

$$y_{n+1} = y_n + \left( \frac{1}{6} \right) [k_1 + 2k_2 + 2k_3 + k_4] \quad (3)$$

Where:

$$k_1 = hf(x_n, y_n)$$

$$k_2 = hf \left( x_n + \frac{h}{2}, y_n + \frac{k_1}{2} \right)$$

$$k_3 = hf \left( x_n + \frac{h}{2}, y_n + \frac{k_2}{2} \right)$$

$$k_4 = hf(x_n + h, y_n + k_3)$$

Here  $h$  is the step size.

### AREA BASED ADAPTIVE REFINEMENT

The model is made manually using Blender 3D modeling software [http://www.blender.org/]. Making a model manually using 3D modeling software, it is not necessary that all triangles area should be the same. Consider Fig. 2 which is a portion of 3D model.

It is clear in Fig. 2, that some triangles have larger area while other has small area. If such mesh is divided by any existing subdivision techniques then all triangles will be

divided irrespective of its area. The disadvantage is that extra triangles need to be rendered and also extra spring force computation and integration.

An area based adaptive subdivision technique is proposed which divides only the large area triangles. Each triangle is divided using Mid Point subdivision technique into 4 triangles during each subdivision. This is shown in Fig. 3. For such subdivision, a threshold area (thresholdArea) is used. The threshold area is used to stop the subdivision and is chosen by hit and trial method during several visual experiments.

During a pre-processing step, the area of each triangle is computed and stored.

#### The pseudo-code of our algorithm

- 1 Detect collision between a Haptic Contact Point (HCP) and triangle Tri in a model
- 2 if (area of Tri > thresholdArea)  
Subdivide (Tri)  
else  
Draw (Tri)
- 3 Find Neighbors of Tri (neighOfTri)
- 4 Loop through All neighOfTri  
While (area of neighOfTri > thresholdArea)  
Subdivide (neighOfTri)  
else  
Draw (neighOfTri)
- 5 end

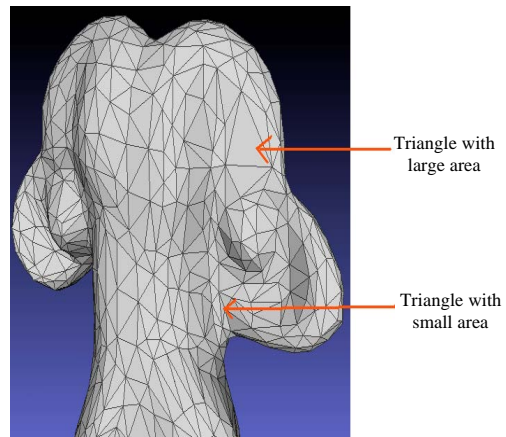


Fig. 2: Coarse triangular mesh of varied triangle areas

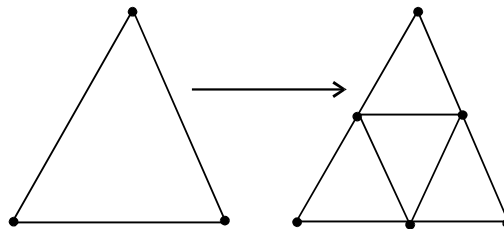


Fig. 3: Mid point subdivision

**SETTING PARAMETERS OF MASS SPRING SYSTEM**

**Setting mass of a new node:** In adaptive subdivision, new points are added which needs to be assigned masses such that it balances the forces applied to the coarse triangular mesh.

The mass of newly added point is calculated as:

$$M_{new} = \frac{\sum_{i=0}^N M_{old}}{numOldM + numNewM} \quad (4)$$

where,  $M_{old}$  point is mass in the coarse mesh and  $M_{new}$  is node in the new subdivided mesh. The numOldM is the number of nodes in the contacted region and numNewM is number of new nodes in the same contacted region through Haptic Contact Point.

**Setting spring constant:** In adaptive refinement the spring constant  $k_s$  needs to be recalculated so that the surface reflects the same force through haptic force feedback as before subdivision, otherwise the surface will feel stiffer. In our algorithm, when the haptic force feedback contact a triangle then this triangle and all its neighbors are subdivided meeting the area condition. So all new nodes must be assigned new spring constant so that it requires the same force for deformation as before and same force reflection  $k'_s$ :

$$k'_s = \frac{k_s}{N} \quad (5)$$

where, N is the number of nodes.

**EXPERIMENTAL RESULTS**

We used several coarse meshes and applied the proposed area based subdivision algorithm as well as the simple subdivision algorithm. It is shown in Fig. 4.

A coarse mesh is taken and is refined using two levels of mid point subdivision and by proposed area based adaptive subdivision methods. Red dot in the Fig. 4 is a node contacted by a haptic force feedback device. In both cases all neighbors which are directly connected to the Node contacted are chosen for subdivision. Midpoint subdivision divides the region without considering the area of a triangle and so small area triangles are refined more. Our proposed algorithm divides only the triangles whose area is under the threshold and thus gives uniform area triangles which during deformation give realistic view.

Figure 5 shows that for the same coarse mesh, to achieve same visual deformation, midpoint algorithm

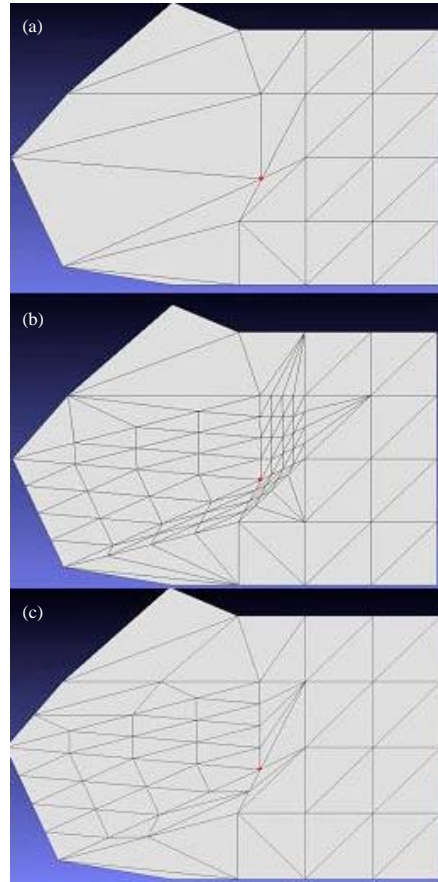


Fig. 4(a-c): (a) Coarse mesh (b) refined using mid point subdivision and (c) refined using area based adaptive subdivision

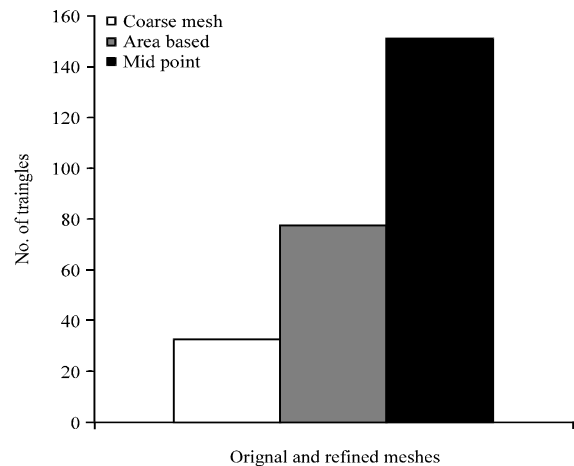


Fig. 5: No. of triangles for the same coarse mesh using mid point and area based adaptive subdivision

produce more triangles as compared to the proposed algorithm. The proposed algorithm is efficient in terms of time as well as memory.

### CONCLUSION

In this study, an area based adaptive subdivision algorithm is proposed and is compared with mid point uniform subdivision algorithm. The algorithm achieves same visual results as mid point uniform algorithm but saves time as well as memory. In haptic environment as the sense response is much higher than visual response, therefore efficient algorithms for deformation are required to meet real time requirement. This algorithm works well for our deformable model because the penetration depth of the haptic probe is not high.

### REFERENCES

- Catmull, E. and J. Clark, 1978. Recursively generated B-spline surfaces on arbitrary topological meshes. *Comput.-Aided Des.*, 10: 350-355.
- Choi, Y.J., M. Hong, M.H. Choi and M.H. Kim, 2002. Adaptive mass-spring simulation using surface wavelet. *Proceedings of the 8th International Conference on Virtual System and MultiMedia*, Sept. 25-27, Gyeong-Ju, Korea, pp: 273-281.
- Doo, D. and M. Sabin, 1978. Behaviour of recursive division surfaces near extraordinary points. *Comput.-Aided Des.*, 10: 356-360.
- Dyn, N., D. Levin and J.A. Gregory, 1990. A butterfly subdivision scheme for surface interpolation with tension control. *ACM. Trans. Graph.*, 9: 160-169.
- Hutchinson, D., M. Preston and T. Hewitt, 1996. Adaptive refinement for mass/spring simulations. *Proceedings of the Eurographics Workshop on Computer Animation and Simulation*, Aug. 31-Sept. 1, Springer-Verlag, New York, pp: 31-45.
- Kahler, K., J. Haber and H.P. Seidel, 2003. Dynamically refining animated triangle meshes for rendering, *The Visual Comput.*, 19: 310-318.
- Kawai, H., 2001. Elastic object manipulation using coarse-to-fine representation of mass-spring models. *Proceedings of the Conference on Abstracts and Applications*.
- Kobbelt, L., 1996. Interpolatory subdivision on open quadrilateral nets with arbitrary topology. *Comput. Graphics Forum*, 15: 409-420.
- Kobbelt, L., 2000. v3-subdivision. *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, July 23-28, New Orleans, LA., USA., pp: 103-112.
- Muller, H. and R. Jaeschke, 1998. Adaptive subdivision curves and surfaces. *Proceedings of the International Conference on Computer Graphics*, June 22-26, Hannover, Germany, pp: 48-58.
- O'Brien, J.F. and J.K. Hodgins, 1999. Graphical modeling and animation of brittle fracture. *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, Aug. 8-13, Los Angeles, CA., USA., pp: 137-147.
- Peterlik, I., M. Sedef, C. Basdogan and L. Matyska, 2010. Real-time visio-haptic interaction with static soft tissue models having geometric and material nonlinearity. *Comput. Graphics*, 34: 43-54.
- Xu, Z. and K. Kondo, 1999. Adaptive refinements in subdivision surfaces. *EuroGraphics'99, Short Study and Demos*, pp: 239-242.
- Zhang, J., S. Payandeh and J. Dill, 2002. Haptic subdivision: An approach to defining level-of-detail in haptic rendering. *Proceedings of the 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, March 24-25, Orlando, FL., pp: 201-208.
- Zorin, D., P. Schroder and W. Sweldens, 1996. Interpolating subdivision for meshes with arbitrary topology. *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, Aug. 4-9, New Orleans, LA., USA., pp: 189-192.