Resolution Scaling for Mass Spring Model Simulations

Maciej KOT\textsuperscript{[a]}, Nonmember, Hiroshi NAGAHASHI\textsuperscript{[b]}, Member, and Krzysztof GRACKI\textsuperscript{[b]}, Nonmember

SUMMARY The volumetric representations of deformable objects suffer from high memory and computational costs. In this work we analyze an approach of constructing low-resolution mass spring models (MSMs) on the basis of a high-resolution reference MSM. Preserving the physical properties of the modeled objects is emphasized such that their motion is consistent and independent of the spring network resolution. We varied the node merging algorithm and analyzed how various aspects of the simplification process affected the properties of the model and how these properties translated into visual behavior in a simulation.

\textit{key words:} level of detail, mass spring model, soft body deformation

1. Introduction

For the purpose of computer-generated animation, the movement of an elastic object and the plausibility of its behavior, which includes the trajectory of its movement as well as the deformations it undergoes, will affect the overall quality of the animation. A plausible behavior can be achieved in a number of ways\cite{1}, \cite{2}. Physically-based approaches of modeling deformable objects, among several such approaches, tend to be the most automatic. They perform quite well in various situations and ideally do not require per-object attention of the programmer or the artist. Finite Element Method (FEM) and Mass Spring Models (MSMs) are two popular examples of techniques from this group.

Computational costs associated with the simulation of soft bodies are significantly higher than they are for rigid bodies. In numerous situations, the internal structure of an object affects its overall behavior; therefore such objects require complex volumetric representations. When presenting large scenes with several objects, it may be beneficial to utilize model reduction mechanisms, similar to level-of-detail (LOD) techniques, which are used for visual representations.

In this article we consider the problem of creating simplified representations of given models. For the representation of deformable objects we chose MSMs because of their simplicity. Our goal is to generate low-resolution MSMs on the basis of high-resolution reference MSMs in a manner that would preserve their physical properties and whose motion would not significantly diverge from the motion of the base model. In many aspects, the problem is analogous to LOD generation for meshes\cite{13}. We adapted known simplification methods such that they could be applied to volumetric representations and preserve not only the shape but also the physical properties.

The underlying motivation behind this study is the reduction of computational costs required to generate an animation of deformable objects. For real time simulations, such as those present in games, reduction in the computational cost may allow the incorporation of higher number of objects in any given scene. For highly complex offline generated animations, the reduced cost may lead to reduction in time required to generate the animation.

Compared with FEM, MSMs tend to be easier to implement, which has made them highly popular; however, MSMs are believed to be less accurate in terms of their physical properties. FEM is used in a majority of applications where either physical correctness or clear error handling is required. Consequently, the attention of researchers dealing with model simplification of elastic objects has been focused on FEM based representations\cite{4},\cite{5},\cite{8},\cite{9}. To the best of our knowledge, this is the first study that addresses the simplification of volumetric MSMs.

2. Related Work

Several techniques exist for the reduction of computational time required for the simulation of deformable objects. In general, efficiency can be improved either by changing the simulation technique (time-integration algorithms, time-step control), or by reducing the complexity of the models. Geometric approaches, such as shape matching\cite{10}, simplify both the models and the simulation technique.

In this study, we focus on reducing the complexity of the models, which can be performed in a number of ways. Kryś et al.\cite{6} and Lall et al.\cite{7} reported model reduction techniques in which they limited the space of possible deformations. Wojtan and Turk\cite{3} employed an FEM representation with embedded high-resolution surface mesh. Nesme et al.\cite{5} used an embedding technique based on an octree voxelization that takes into account the non-homogeneity of modeled objects. Numerous popular approaches are based on adaptive model refinement, especially for FEM-based simulations. In these approaches, the initial low levels of detail in an object representation are refined when required.
to achieve better accuracy. Budd et al. [26] provided a survey of moving mesh methods. Mesh refinement can be used for both clothes and volumetric objects. Examples of the first group include Provot [19] for two-dimensional (2D) MSM meshes or Narain et al. [18] for FEM representations. More information regarding surface remeshing can be found in the survey by Alliez et al. [20]. Examples of using mesh refinement for volumetric objects include Debuine et al. [9] or Wicke et al. [8]. In this study, we also consider three-dimensional (3D) volumetric objects; however, our techniques are not adaptive. The main idea behind adaptive remeshing is to add detail, only when necessary, in certain areas of the object as the simulation progresses. This is certainly one of the most efficient ways of representing simulated objects. However, it is worthwhile to note that after a similar evolution of LOD techniques for visualization, adaptive methods have never become a standard in the game industry. Instead, non-adaptive methods are widely used in practical applications because of their lower implantation complexity. However, there are situations where non-adaptive methods cannot be applied such as elasto-plastic deformations (Wicke et al. [8]).

For FEM representations, the problem of model simplification that is most similar to ours has been studied by Kharevych et al. [4]. They formulated elasticity on coarse resolutions by deriving effective elasticity tensors. An advantage of their method is the ability to handle heterogeneous objects. MSMs are very popular in cloth modeling; however, relatively few researchers have considered real volumetric models for 3D objects. Therefore, we aimed to implement a method for the simplification of MSMs. A disadvantage of the MSMs is the difficulty in defining physical properties of the spring network. Although several attempts were made to identify MSM parameters of particular materials, most of them dealt with very specific cases of materials and required additional modifications to mass-spring systems such as adding custom constraints or “angular springs.” The basic and general case that we want to analyze is an MSM composed of mass points and linear springs. The description of such an MSM was given by Lloyd et al. [15]; however, their equations are limited to models based on tetrahedral meshes. They obtained spring parameters by comparing MSM to a corresponding FEM-based model. An explicit equation was provided for spring coefficients for regular tetrahedra. The description of an MSM with mass points distributed on a cubic lattice was given by Ladd and Kinney [14]. They derived spring constants from linear elasticity equations. In addition, a more physically-oriented approach to the different types of MSMs along with the techniques for the estimation of their physical properties can be found in the review article by Ostoja-Starzewski [11].

Another concern with model creation and simplification is judging the quality of a model and the quality of the resulting animation. It is difficult to decide which behavioral artifacts are acceptable and unlikely to be noticed. Yeh et al. [16] provided an overview of this topic and explored a few approaches of identifying the error tolerances of physical simulation. They proposed the use of maximum percentage energy differences to evaluate the perceptual quality of the simulation.

3. Properties of Mass-Spring Models

One of the most natural and commonly used MSM types is based on the distribution of mass points on a cubic lattice, which we refer to as cubic lattice MSM or cubic MSM. Typically, spring creation rules assume that the closest and the next-closest neighbors are connected by springs with the same spring constant $k$. Springs connecting the closest neighbors are placed along the cube edges, whereas the next-closest are placed along the diagonals of cube faces. In such a case, the elastic properties of the resulting objects can be obtained from linear elasticity equations [14] and are given by:

$$E = 2.5 \frac{k}{a}, \quad \nu = 0.25,$$

where $E$ is Young’s modulus, $\nu$ is Poisson’s ratio, and $a$ is the length of an elementary cube in the lattice. In fact cubic lattice MSMs correctly represent homogeneous isotropic material, which we verified in our previous work [21]. We use them as high-resolution reference models for the simplification algorithm.

A more general description of an MSM can be obtained from its statistical properties. For a uniform compression of an MSM, resulting in a deformation in which the strain is equal $\epsilon$ in every direction, the change in the energy density associated with the deformation is given by

$$\Delta e = \frac{1}{2} \sum_{i} k_i \epsilon^2 \frac{L_i^2}{V},$$

where the sum is calculated considering all the springs inside a volume $V$, $k_i$ is the spring coefficient, and $L_i$ is the natural length of $i$-th spring. From the basic elasticity theory [12] $\Delta e$ in the uniform compression is given by

$$\Delta e = \frac{9}{2} K \epsilon^2,$$

where $K$ is a bulk modulus. Young’s modulus can be expressed by $K$ and $\nu$ as:

$$E = 3K(1-2\nu), \quad \nu = 0.25,$$

which gives

$$E = \frac{1}{6V} \sum_{i} k_i L_i^2 \approx \frac{N S}{12V} \bar{k} L^2$$

as an approximation of Young’s modulus for a given volume $V$ of a material represented by an MSM with $N$ nodes connected by springs with spring coefficient $\bar{k}$. The average number of springs attached to a node is equal to $S$ and the average length of a spring is equal to $L$.
It is important to note that the MSMs with random node placement are very flexible, they allow the use of arbitrary spring connection rules as well as node positioning. Equation (5) is approximate and the more homogeneous and isotropic the MSM is, the better this approximation is expected to work. In fact, our tests demonstrate that this approximation works very well for models having 14 or more springs attached to one node with reasonably uniform mass point distributions. It can be applied to smaller parts of the object as well as to non-homogeneous objects giving averaged properties over considered volumes.

Both cubic lattice MSM and random MSM allow the construction of objects composed of isotropic materials with Poisson’s ratio equal to 0.25, which cannot be changed. Such fixed Poisson’s ratio is the limitation of models based on central forces only. When considering applications such as games or animations, this parameter is fortunately less important. Plausible models can be created just by choosing an appropriate Young’s modulus.

4. Model Simplification

When creating an animation, very precise and detailed models are not always necessary. In certain cases, it is desirable to generate a simplified version of an object or at the very least one that is not overly detailed. The level of detail in the description determines the scale of physical phenomena, which an object can undergo without damaging the accuracy of the simulation. Only very simple deformations will be possible if the level of detail is greatly reduced.

4.1 Global Resolution Scaling

The simplest approach for constructing a lower resolution model of an object is to fill its volume with the desired number of mass points, on the basis of either a cubic lattice distribution or a random distribution. An example is shown in Fig. 1. Spring coefficients are chosen according to Eqs. (1) or (5). To reflect local changes in the material strength, both Eqs. (1) and (5) can be locally applied to small portions of an object. In general, the global resolution scaling is analogous to global simplification methods of meshes with all their advantages and drawbacks. Its most noticeable characteristic is that different representations of the object generated in this manner are in principle not directly related to each other, which makes the problem of defining a clear correspondence between such representations more challenging. This can be important for switching the object representation during simulation; however, we do not analyze this aspect in the current study. Global scaling is best applied to homogeneous models with simple geometry. For complex objects constructed of heterogeneous materials, local simplification techniques are more suitable because they allow a better control of local features. In these approaches, not every part of the object have to be simplified to the same extent.

4.2 Simplification by Node Merging

A class of popular LOD generation algorithms is based on local simplification operators, which are capable of reducing details in a given area of an object. Such operators would be applied throughout the mesh or the volume, resulting in its simplification. Methods based on local simplification operators achieve better consistency between models with different resolutions because simpler representations are based on highly detailed representations. In classical LODs for meshes, local simplification operators include vertex removal, edge collapse, triangle collapse, or vertex clustering. The basic ideas behind edge collapse, triangle collapse, and vertex collapse are very similar, i.e., multiple primitives are merged into one. We adapted this concept for 3D mass-spring systems.

The procedure of merging multiple nodes of the mass-spring system into one is the core of the analyzed approach. The conditions of merging and the problem of choosing which nodes should be merged are discussed in the next section. In this section, we focus on the merging step. Our goal is to replace two or more mass points with a new point. The merging algorithm performs the following procedure:

- Set the mass of the new point equal to the sum of the masses of previous points.
- Set the position to the center of mass of the previous points.
- If a previous node has been connected by a spring with any other node, the new node will also be connected (all previous springs are reattached to the new node). Each spring affected by this procedure changes its coefficient $k$ proportionally to the inverse of its length $L$ (following the rule of series $\frac{1}{k_{new}} = \frac{1}{k_1} + \frac{1}{k_2}$):

$$\frac{k}{k_0} = \frac{L_0}{L},$$

Fig. 1 A cube represented by cubic lattice MSM with four different resolutions. Number of nodes: 27, 64, 125, 216.
where $k$ and $L$ are the new values, and $k_0, L_0$ are the previous ones.

- Parallel spring connections are replaced by a single spring with appropriate $k$ (by the rule of parallel connections, $k_{eff} = k_A + k_B$).
- All damping coefficients change proportionally with $k$ [17].

Using this approach, springs between nodes that undergo merging are discarded; therefore, the local properties of the material should not depend on singular springs but should be reflected by the coefficients of numerous springs. This can be achieved by any reasonable technique of constructing an MSM that does not place local properties of the material on one spring alone but distributes them among several springs.

The first condition (sum of masses) implies that we assume that the volume represented by a point is proportional to its mass. It is difficult to define the exact volume represented by each mass point, as it depends on the topology of the mass-spring system. We use the following approximation. We treat each node $i$ as a spherical object with a radius equal $L_i/2$, where $L_i$ is the average length of springs connected to the $i$-th node. This approximation is sufficient for surface representation as well as collision detection.

Alternatively, the closest proximity representation can be used, where each point of a continuous object is represented by the closest node from a discretized model. In such a case, after merging, masses will also be distributed between neighboring nodes that did not directly participate in the merging process. The difference between the two approaches is shown in Fig. 2. Both are equivalent in terms of static properties of an object; however, choosing one over another will slightly affect the dynamics. In our tests, we used the first approach because of its simplicity. Note that the simplification procedure slightly changes the moment of inertia of the object. This effect is reduced when the simplification is applied uniformly throughout the entire object.

5. Node Grouping for the Simplification

In our tests, we considered a few variations of the node-merging approach. The first approach is a simple merging of closest neighbors. More sophisticated techniques take into account the heterogeneity of the object, as well as try to preserve the border of the object. In all the algorithms, we only merge vertices connected by springs, which eliminates the potential problem of merging two mechanically separate parts of the object.

5.1 Standard Merging

Using standard merging, groups of two or three mass points are chosen as candidates for merging on the basis of geometrical proximity. The grouping is achieved using the following recursive clustering procedure. In each step, a set of mass points is divided into two groups each of which is further divided until the number of points in a group is lower than four. Each time, the point furthest away from the center of mass of a current set is chosen as a seed of the first group and another, furthest away from the first point, as a seed of the second group. All the other points from the set are classified as belonging to Group 1 or Group 2 on the basis of the distance from the seed points. In addition, after each step, for every group the center of mass is computed and for each point the distances to closest centers are calculated. A point is transferred if it is closer to the center of mass of another group than it is to its own group’s center. A group is merged with its closest group if it contains fewer than two points. A group is divided in two subgroups if it contains more than three points. The algorithm stops when the global number of 2- and 3-point groups stops increasing.

The examples of LOD models of a cube generated by this technique are shown in Fig. 3. The disadvantage of this method is, that it does not take into account heterogeneity. When the composites of different materials are simplified, high gradients of mass density or stiffness in the object are smoothed out by this procedure. In addition, the shape of the object might dramatically change, which is unacceptable for numerous models.

5.2 Priority Merging

To preserve the heterogeneity of the object, we introduced a modified distance function $d_{ij}$ into the merging, which is
used in the grouping procedure instead of the geometric distance $d_{ij}$.

$$d_{ij}^* = d_{ij} \cdot [1 + w \cdot (|\nabla E| + |\nabla m|)^z]. \quad (7)$$

The default values of both the parameters $w$ and $z$ are 1; however, they are adjustable and change how $d_{ij}^*$ diverges from the standard distance measure.

With the use of $d_{ij}^*$, homogeneous areas are simplified first, which results in a better preservation of the physical properties of more complex objects. Quality improvement of LODs generated with this algorithm is noticeable, especially when multiple simplifications are performed with the same model. The distance function that we use computes the gradient of Young’s modulus $E$ (estimated with Eq. (5)) along with the gradient of mass density inside the object. Areas with small gradients are given higher priorities for merging.

5.3 Merging with Border Preservation

A modification of the node-merging algorithms allows a reduction of the effect of shape degradation caused by simplification. We distinguish two types of border preservation.

5.3.1 Strong Preservation

The strong preservation method prevents the merging of the points that lie on the border. This method produces no simplifications of the outer areas of the object; therefore, the shape of the original model remains the same (Fig. 4). In certain situations, this will be important for object-object interaction, because the border is the area of first contact. At the same time, it imposes very serious limitations on simplification algorithms. For small objects, the border area is relatively big compared to the volume, and if we prevent simplification, there is not much left for the algorithm to work with (i.e., cube modeled by $5 \times 5 \times 5$ lattice points consists of 125 nodes, 98 of which lie on the surface). It is important to note that if the forces involved in an interaction are big, the shape of the object will not play a dominant role; therefore, such a strict border preservation method might not improve the overall behavior very much.

5.3.2 Weak Preservation

Weak preservation is another lenient approach for shape preservation that imposes the following rule on the node-merging operator: nodes that lie on the border can be merged only with each other. Figure 5 shows a model of a rabbit simplified using this procedure. For any border node, if the best candidate for merging is a node that lies inside the object, it will not be merged at all (the algorithm will not try to find another candidate that might be further away yet satisfies the condition of being on the border as well). This prevents points that lie inside the object from becoming border points because of any merging operation. Narrow and high surface curvature areas are preserved, whereas big and flat areas are simplified.

6. Tests

In this section, we use the following notation:

- LOD-n: n-th level of simplification (each simplification reduces the number of nodes by about 50%)
- LOD-h-n: priority merging
- LOD-b-n: weak border preservation. Combinations of any of the above are possible.

We do not show tests for LODs generated with strong border preservation. Although they allow for a better preservation of physical properties, the strong limitation on the simplification procedure makes it impossible to achieve a desirable reduction in the number of nodes for the most of the models.

6.1 Elastic Parameters

To test the performance of generated LODs, first we numer-
A force is applied to the rabbit’s head, pulling it backwards. From the left: initial non-deformed hi-res model (2479 nodes). Next, resulting deformation of the hi-res model, LOD-b-1 (1085 nodes) and LOD-b-2 (482 nodes) models.

Fig. 7 Energy dissipation test. Left: object falls on the ground. Right: object is rolling.

Dense core with soft shell. From the left: cubic lattice MSM model, $N = 281$, $S = 4094$, LOD-hb-1: $N = 114$, $S = 1596$, LOD-hb-2: $N = 51$, $S = 670$. Numerically verified that the elastic parameters of the MSM do not significantly change between different LODs. During this test, we measured the response of a big uniform cuboid object to a compression in $x$ direction. The relative change in cuboid length, in each direction, was estimated by measuring the displacements of the points located close to the border (within $0.1L_y$), and averaging them. Poisson’s ratio was calculated as $\nu = -\epsilon_y/\epsilon_x$. Next, we measured the stress within the object with Hardy’s method [27], and calculated $E = \sigma_{xx} - \nu \cdot (\sigma_{yy} + \sigma_{zz})/\epsilon_x$, where $\sigma$ denotes stress tensor and $\epsilon$ denotes strain.

The original object was composed of 29k nodes, the first LOD comprised 13k nodes, and the second LOD comprised 6k nodes. The measured values of Young’s modulus and Poisson’s ratio were:

$E : 100.67, 99.87, 110.7$;
$\nu : 0.255, 0.266, 0.28$.

As the results show, the values are quite well preserved. The most simplified version of the object diverges from its original elastic properties only by about 10%. To see how these results translate into the real behavior of a sample object, we applied a force to the head of an elastic bunny and observed the resulting deformation for three levels of detail. The results can be seen in Fig. 6 and they are very satisfactory.

Table 1 The distance a spherical object travels in Scenario 1 before it stops.

| model       | distance | difference | nodes | springs |
|-------------|----------|------------|-------|---------|
| Base model  | 6.1      | 0          | 281   | 4094    |
| LOD-1       | 4.5      | 1.6        | 123   | 1762    |
| LOD-b-1     | 4.6      | 1.5        | 121   | 1784    |
| LOD-2       | 5        | 1.1        | 55    | 722     |
| LOD-b-2     | 4.5      | 1.6        | 53    | 780     |
| LOD-h-1     | 5.6      | 0.5        | 114   | 1596    |
| LOD-hb-1    | 5.6      | 0.5        | 114   | 1596    |
| LOD-h-2     | 5.5      | 0.6        | 51    | 670     |
| LOD-hb-2    | 6        | 0.1        | 51    | 670     |

Next, we performed two tests in which the behavior of the object is driven by two different physical phenomena.

6.2 Scenario 1 – Energy Dissipation

In the first scenario a deformable spherical object with a dense core and a soft shell is horizontally thrown just above a flat surface. It falls down and after a small bounce starts rolling until it stops because of energy dissipation that occurs mostly in the soft shell (Figs. 7 and 8). The original object with mass $M$ is constructed using a cubic lattice MSM...
with lattice spacing $a_0$. The unit of force is defined as $[k_0 a_0]$. 
The radius of the core $R_c$ is $2a_0$ and the radius of the whole object $R$ is $6a_0$. The mass of the core $m_c$ is $0.53M$. Young’s modulus of the core $E$ is $125[k_0 a_0]$, which corresponds to the spring coefficients $k = 12.5k_0$. The shell is 10 times weaker ($E = 12.5[k_0 a_0]$). The damping coefficients of the springs in the core and the shell are set to $0.05[k_0 a_0]$, where $t_0$ is a unit of time and the friction coefficient of the surface is set to 0.3.

All the parameters were chosen by a trial and error method to achieve a plausible behavior, and they do not correspond to any actual values of real materials. The simplification parameters for Eq. (7) were set to $w = 1$ and $z = 1$.

For each LOD, we measured and compared the distance travelled by them. The travelled distance highly depends on the properties of the shell, as the overall movement is quite long, and all the behavioral differences are accumu-
lating with a snowball effect. The results along with the information regarding the number of mass points and springs utilized for object representation are presented in Table 1. As the table shows, LODs generated with priority merging behave better than those generated with standard merging. Surprisingly, enabling border preservation does not change the behavior of an object in a significant manner. Originally, this scenario had been specifically designed for testing the influence of border preservation on the quality of the model because the behavior of this object is driven by deformations that occur in its outer regions. The visual quality of the animation is satisfactory for all generated LODs, and the most simplified ones have about 18% of the original number of nodes and springs.

6.3 Scenario 2 – Complex Movement

In the second test case, an object composed of two very stiff rods connected by a rubber is thrown at the wall while rotating (Figs. 9 and 10). It hits the wall, bounces off, and falls to the ground while possibly still rotating until it finally stops because of friction. The exact movement is quite complex and the trajectory heavily depends on the initial velocity of the object. In this test, we used models generated by the standard and priority merging algorithms, both with weak border preservation. The entire object had a mass $M$, 87% of which was attributed to the stiff parts. The original object was constructed with a cubic lattice MSM with a lattice spacing at 0.25$a_0$ and its dimensions were 6$a_0 \times a_0 \times a_0$. The rubber part was placed precisely in the middle and its dimensions were $a_0 \times a_0 \times a_0$. Young’s modulus $E$ of the stiff parts was 2800$[\frac{E}{a_0}]$, which corresponded to the value of spring coefficients $k$ of 700$k_0$. The Young’s modulus $E$ of the elastic part was 140$[\frac{E}{a_0}]$, which corresponded to $k$ of 35$k_0$. In both cases damping coefficients were equal to 0.3$[k_0 t_0]$. Friction coefficient of the ground was set to 0.3. Initial velocity of the right stiff part was set to (10, 20, 33)$[a_0 t_0]$, and the gravity was (0, −10, 0)$[a_0 t_0^2]$. The bar was placed at 4$a_0$ above the ground. All the parameters were chosen by a trial and error method to achieve a plausible behavior, and they did not correspond to any actual values of real materials. The simplification parameters for Eq. (7) were set to $w = 1$ and $z = 1$.

We compared the final position of the center of mass of the object among different LODs. Each position was measured as a distance from the wall. The simulation was performed with a variable time step and it was not deterministic. Every position presented in the table was taken as an average from three runs of the simulation for each model. We chose to ignore the final orientation of the bar because its distribution was too chaotic and might have significantly changed even between different runs of the same model. Figure 9 shows the original model and an example of a simplified one. Snapshots of the animation are shown in Fig. 10.

Results of this test are presented in Table 2. They confirm that preserving the non-homogeneous areas of the object improved the quality of simplified representations. LODs generated with priority merging exhibited a behavior closer to the original. Both first level LODs (composed of about 45% of the base number of nodes and springs) allowed the animation to recreate the original movement of the object surprisingly well. Trajectories of the second level LODs (20% of the original complexity) were noticeably different; however, the behavior of these LODs was still plausible.

7. Conclusions

In this work, we proposed a methodology for the simplification of mass-spring model representations of deformable objects. Our tests indicated that the proposed adaptation of a classical LOD merging approach allowed us to obtain very good results with volumetric MSM models. We compared a few variations of the presented algorithm, including merging with border and heterogeneity preservation. Our tests indicated that preserving the heterogeneous areas had a noticeable positive effect on the quality of simplified models. The border preservation was expected to affect different objects in a different manner, depending on the object’s shape. However, we did not observe significant behavioral changes caused by enabling or disabling this modification for any of our test schemes.

In our tests, objects simplified by as much as 50% exhibited very small differences from the original behavior. Further simplification led to noticeable changes in elastic properties; however, in many cases, the behavior of such objects was still plausible. There is a limitation regarding the extend of the reduction in the resolution of a model. Subsequent simplifications change the network topology and could cause certain regions of the network to rapidly loose their elastic properties because of insufficient network connectivity. Therefore, one way to improve our method concerning this point would be to impose a control over the number of springs attached to each node and their angular distribution. Moreover, it should be noted, that mass spring models are based on linear elasticity theory, which has a serious limitation. Such models are expected to provide a plausible elastic response only within its limits, i.e., for small relative displacements.

Our technique was verified with four different experiments. Simple compression showed that the elastic parameters of a homogeneous isotropic material were quite well preserved with the simplification, and the study of the deformation of a rabbit confirmed that it translates into a good visual behavior for non-trivial shapes. Energy dissipation and complex movement tests showed, that heterogeneous models can also be handled by our technique.

The results of these experiments are encouraging, but our tests were by no means exhaustive. We did not verify the the effects of simplification on more exotic materials such as those exhibiting anisotropy, having a smooth distribution of the stiffness, or even varying resolution of the mass spring network. However, there are no indications which would suggest that our method cannot handle such models. The
advantage of our technique is its simplicity because it is not based on a sophisticated analysis of the network but rather operates on the simplest elements with straightforward rules that make no assumptions about the properties of the material.

References

[1] S.F.F. Gibson and B. Mirtich, “A survey of deformable models in computer graphics,” Technical Report TR-97-19, MERL, Cambridge, MA, 1997.

[2] A. Nealen, M. Müller, R. Keiser, E. Boxerman, and M. Carlson, “Physically based deformable models in computer graphics,” Comput. Graph. Forum, vol.25, no.4, pp.809–836, 2006.

[3] C. Wojtan and G. Turk, “Fast viscoelastic behavior with thin features,” ACM Trans. Graphics, vol.27, no.3, Article 47, Aug. 2008.

[4] L. Kharevych, P. Mullen, H. Owhadi, and M. Desbrun, “Numerical coarsening of inhomogeneous elastic materials,” ACM Trans. Graphics, vol.28, no.3, Article 51, 2009.

[5] M. Nesme, P.G. Kry, L. Jefábková, and F. Faure, “Preserving topology and elasticity for embedded deformable models,” ACM Trans. Graphics, vol.28, no.3, Article 52, 2009.

[6] P. Krysl, S. Lall, and J.E. Marsden, “Dimensional model reduction in non-linear finite element dynamics of solids and structures;” Int. J. Numer. Meth. Engng, vol.51, pp.479–504, 2001.

[7] S. Lall, P. Krysl, and J.E. Marsden, “Structure-preserving model reduction for mechanical systems,” Physica D, vol.184, no.1-4, pp.304–318, 2003.

[8] M. Wicke, D. Ritchie, B.M. Klingner, S. Burke, J.R. Shewchuk, and J.F. O’Brien, “Dynamic local remeshing for elastoplastic simulation,” ACM Trans. Graph., vol.29, pp.49:1–49:11, July 2010.

[9] G. Debunne, M. Desbrun, M.-P. Cani, and A.H. Barr, “Dynamic real-time deformations using space & time adaptive sampling,” Proc. 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ’01, pp.31–36, 2001.

[10] M. Müller, B. Heidelberger, M. Teschner, and M. Gross, “Meshless Deformations Based on Shape Matching,” ACM Trans. Graphics, vol.24, no.3, pp.471–478, Aug. 2005.

[11] M. Ostoja-Starzewski, “Lattice models in micromechanics;” Appl. Mech. Rev., vol.55, pp.35–61, 2002.

[12] L.D. Landau and E.M. Lifshitz, Theory of elasticity, Prgamon, London, 1959.

[13] D. Luebke, M. Reddy, J. Cohen, A. Varshney, B. Watson, and R. Huebner, Level of Detail for 3D Graphics, Morgan Kaufmann, 2002.

[14] A.J.C. Ladd and J.H. Kinney, “Elastic constants of cellular structures,” Physica A, vol.240, pp.349–360, 1997.

[15] B.A. Lloyd, G. Szekeley, and M. Harders, “Identification of spring parameters for deformable object simulation;” IEEE Trans. Vis. Comput. Graph., vol.13, no.5, pp.1081–1094, 2007.

[16] T.Y. Yeh, G. Reinman, S.J. Patel, and P. Faloutsos, “Fool Me Twice: Exploring and exploiting error tolerance in physics-based animation;” ACM Trans. Graphics, vol.29, no.1, Article 5, 2009.

[17] B.A. Lloyd, S. Kirac, G. Szekeley, and M. Harders, “Identification of dynamic mass spring parameters for deformable body simulation;” Eurographics 2008 - Short Papers, pp.131–134, 2008.

[18] R. Narain, A. Samii, and J.F. O’Brien, “Adaptive anisotropic remeshing for cloth simulation;” ACM Trans. Graphics, vol.31, pp.147:1–147:10, Nov. 2012.

[19] X. Provot, “Deformation constraints in a Mass-Spring Model to describe rigid cloth behavior;” Graphics Interface, pp.147–154, 1996.

[20] P. Alliez, M. Attene, C. Gotsman, and G. Ucelli, “Recent advances in remeshing of surfaces,” in Shape Analysis and Structuring, ch.2, Springer, 2007.

[21] M. Kot, H. Nagahashi, and P. Szymczak, “Verification of physical properties of materials modeled with Mass-Spring Systems,” IEICE Technical Report, IE2010-180, March 2011.

[22] S. Pabst, B. Thomaszewski, and W. Strasser, “Anisotropic friction for deformable surfaces and solids,” Proc. 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp.149–154, 2009.

[23] R. Brideon, R. Fedkiw, and J. Anderson, “Robust treatment of collisions, contact and friction for cloth animation;” SIGGRAPH’02, pp.43–54, 2002.

[24] M. Kot, H. Nagahashi, and K. Krakli, “Simplification of mass spring models,” Proc. 2012 International Conference on Computer Graphics & Virtual Reality, July 2012.

[25] S. Capell, S. Green, B. Curless, T. Duchamp, and Z. Popovic, “A multiresolution framework for dynamic deformations;” Proc. Symposium on Computer Animation, pp.41–48, 2002.

[26] C.J. Budd, W. Huang, and R.D. Russell, “Adaptivity with moving grids;” Acta Numerica 2009, vol.18, pp.1–131, 2009.

[27] R.J. Hardy, “Formulas for determining local properties in molecular dynamics simulations: Shock waves,” J. Chem. Phys., vol.76, no.1, pp.622–628, 1982.