The Visual Computer manuscript No.
(will be inserted by the editor)

Young-Min Kang · Burkhard Wünsche · Hwan-Gue Cho

Interactive Simulation of Folded Paper with Breakable Hinge Springs and Fractal Surface

Realtime animation of damage preserving adaptive mesh with stiff springs

Abstract There have been little efforts to animate paper objects in interactive or realtime VR environments. Paper can be represented with the well-known mass-spring model, whose animation is relatively easy compared with rigid body models or fluid simulation. However, animating paper with the straightforward mass-spring model results in a serious numerical problem. The properties of plain paper require the representing mass-spring model to be extremely stiff, and the stiffness causes numerical instability. This numerical instability has been one of major obstacles to the real-time cloth animation, and many approaches have been proposed for efficient and plausible simulation of fabric material such as cloth. However, paper is far stiffer than cloth and in addition often fragile. The paper does not recover its original shape when it is deformed with strong force. The traditional mass-spring system cannot represent this damage preserving properties. In this paper, we propose a simple and intuitive approaches to animate the damage-preserving stiff mass-spring model in order to represent the behavior of paper in virtual environments. The proposed method uses an approximate implicit method to guarantee the stability of the system, and employs breakable hinge springs to show damage-preserving properties. Fractal textures represent the small scale wrinkled appearance of old paper. The proposed method can be successfully utilized in realtime or interactive games or VR environments for paper animation.

Keywords Paper Simulation, Paper Folding, Breakable Hinge Spring, Fractal Surface

Young-Min Kang
Department of Game Engineering, Tongmyong University, Korea *ymkang@tu.ac.kr*

Burkhard Wünsche
Department of Computer Engineering, Auckland University, New Zealand, *bwue001@cs.auckland.ac.nz*

Hwan-Gue Cho
Department of Computer Science & Engineering Pusan National University, Korea *hgcho@pusan.ac.kr*

1 Introduction

Although paper is a common object in everyday life, there have been little efforts to animate paper objects in interactive or realtime VR environments. Recently computational geometry has shown some interests in the combinatorial property of folded paper in the computational origamis[6]. The dynamic behavior of paper has been modeled by Chu et al. in order to simulate page turns for virtual books[4]. The authors use a mass-spring structure defined on a rectangular grid of particles. These are connected with primary mesh springs, diagonal bracing, and four-way flexural rigidity springs. Many of the springs are very stiff, leading to a stiff system of differential equations. The resulting system of differential equations is solved with a Runga-Kutta method which can take several minutes for a single page-turn, precluding real-time operation. Other problems are that the mesh is static, has a low resolution and can only be bend smoothly and with a low curvature. Hence its not possible to model other interactions with the paper such as folding, ripping or crumpling it. Also area preservation is not ensured.

Wrinkled or folded paper models can be used as stylized images to create impressive effects in commercial art work as demonstrated in Fig.1. When we carefully investigate the image of Fig.1, it can be seen that it is not a real photo but instead most likely generated by imposing a text printing over the scanned image of a wrinkled paper. As far as we know there have been few studies how to simulate the paper folding/unfolding mechanism in the view of computer graphics rendering.

In order to represent the behavior of a paper object, the well-known mass-spring model can be easily employed. The mass-spring model is the most intuitive and simple representation of deformable soft objects. In the computer graphics society, the animation of soft objects has been intensively studied for a couple of decades. The animation of soft objects modeled with mass-spring systems is far more time-consuming than that of a rigid object because of numerical instability. One of the most

common applications of mass-spring based soft object models is the efficient simulation of cloth, which is represented with stiff springs connecting mass-points.

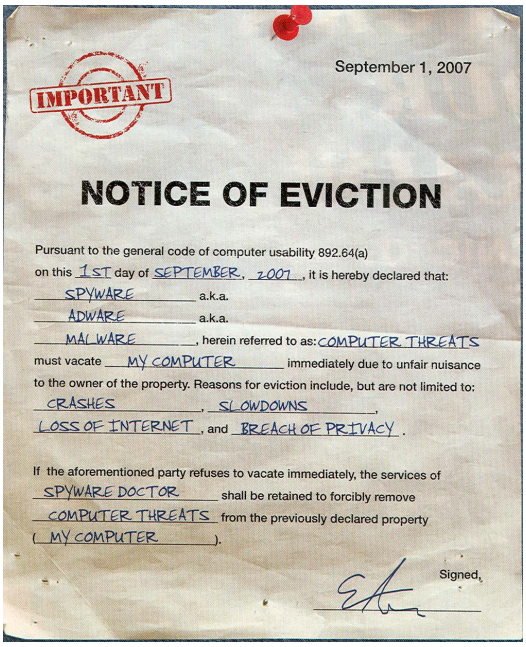


Fig. 1 A slightly wrinkled paper image used in an advertizement page in "WIRED" magazine

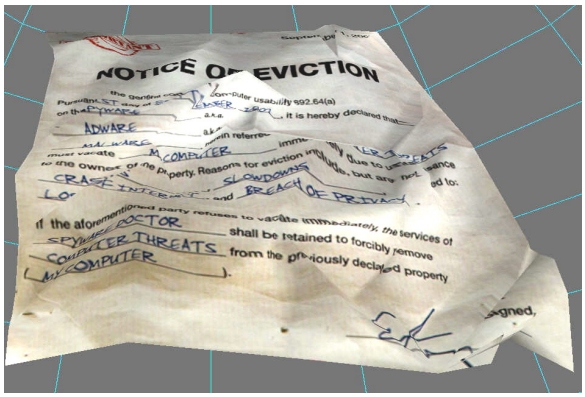


Fig. 2 Virtual paper constructed by this paper for the interactive environments. The mapping texture was the scanned page image in Fig.1

However, animating paper with mass-spring model leads to a serious numerical problem. The properties of paper require the representing mass-spring model to be extremely stiff, and the stiffness causes numerical instability.

In the early stage of deformable object animation, geometric approaches have been widely employed because physical-based models require too heavy computation[10]. Terzopoulos *et al.* characterized the soft object animation as a deformable surface problem [12], and their work formed the foundation for the subsequent physics-based approaches.

Physical-based modeling approaches to soft object animation suffered from numerical instability which was a major obstacle to the real-time animation of stiff deformable objects such as cloth and paper. However, the use of implicit methods has resulted in many approximate approaches for the efficient animation of virtual cloth. Paper is far stiffer than cloth and also more also fragile. The paper does not recover its original shape when it is deformed with strong force. The traditional mass-spring system cannot represent this damage preserving properties.

In this paper, we propose an simple and intuitive approaches to animate the damage-preserving stiff mass-spring model in order to represent the behavior of paper in virtual environments.

2 Problem and Previous Work

The interactive animation of paper can be described as an efficient numerical integration of motion equation for the damage preserving stiff mass-spring model. This problem is composed of two main issues: 1) stable integration of a stiff ODE (ordinary differential equation), and 2) mesh management for damage preservation.

When modeled with a mass-spring model, paper is extremely stiff because paper hardly expands or contracts. Therefore, the animation of paper can be reduced to a numerical integration of extremely stiff ordinary differential equations. In order to animate the paper in virtual world, stable and efficient integration method has to be employed.

Although the stability is an essential issue for paper simulation, stable animation does not guarantee the realism of paper animation. If a paper object is animated with a stable mass-spring simulation method, the resulting animation would look like rubber animation rather than stiff but fragile paper. In order to increase the realism of paper animation, a damage preserving property must be taken into account.

Since Terzopoulos *et al.* represented soft objects as deformable surfaces [12], various physically-based approaches have been proposed for soft object animation. Thalmann's group has proposed many techniques for dressed virtual characters [13]. However, their major interest was the realism of the virtual cloth instead of computational efficiency. Therefore, most of their work employed explicit integration and interactive or realtime animation was impossible.

Various techniques have been proposed for efficient animation of soft objects, and one of the most important advances is the use of implicit integration which guarantees the stability of the animation [1]. However, the implicit integration involves large linear system [5]. Some

efficient approximate approaches have been proposed [9]. A stable method that approximates the solution with a direct update formula was also introduced [7].

3 Interactive Paper Animation Techniques

In this section, techniques for interactive animation of paper objects will be introduced. The techniques are composed of three different approaches for stable animation, damage preservation, and surface details.

In order to obtain realtime or interactive performance, we employed an approximate implicit integration. The method provides stable and efficient computation of the next state of stiff mass-spring model. Our breakable hinge spring model is an approach to the representation of damage preserving paper objects. The model adaptively adjusts the mesh structure of the mass-spring model in order to preserve the damage and express various creases on the paper. The last technique is to generate the surface detail with a fractal surface texture.

3.1 Stable Animation of Extremely Stiff Springs

The stability is essential for the realtime or interactive animation of mass-spring model, and implicit integration is the common solution to the stability problem. The most simple implicit method is the backward Euler method, which can be described as follows:

$$\begin{pmatrix} \mathbf{v}^{t+h} \\ \mathbf{x}^{t+h} \end{pmatrix} = \begin{pmatrix} \mathbf{v}^{t} + h\mathbf{M}^{-1}\mathbf{f}^{t+h} \\ \mathbf{x}^{t} + h\mathbf{v}^{t+h} \end{pmatrix} \tag{1}$$

where h denotes the time interval, $\mathbf{v}$ the vector of velocity values of the mass-points, $\mathbf{f}$ the vector of forces, $\mathbf{x}$ the vector of locations of the mass-points, and $\mathbf{M}$ the mass matrix.

Although the implicit method described in Eq.1 seems simple, real-time animation is not an easy problem. The difficulty arises from the unknown force vector at time $t+h$. Since Hooke's law can compute only $\mathbf{f}^t$, the force at the next time step should be approximated by applying Taylor expansion as follows:

$$\mathbf{f}^{t+h} = \mathbf{f}^t + \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \Delta\mathbf{x}^{t+h} = \mathbf{f}^t + \mathbf{J}\Delta\mathbf{x}^{t+h} \tag{2}$$

where $\mathbf{J}$ denotes the Jacobian matrix of the force vector with respect to the position vector.

By using the force shown in Eq.2, an animation step of the mass spring model can be represented by a typical $\mathbf{A}x = b$ linear system:

$$(\mathbf{M} - h^2\mathbf{J})\Delta\mathbf{v}^{t+h} = h\mathbf{f}^t + h^2\mathbf{J}\mathbf{v}^t \tag{3}$$

For simplicity, let $\mathbf{W}$ denote the matrix $(\mathbf{M} - h^2\mathbf{J})$. The right side of Eq.3 is the sum of a spring force and viscosity force and can be easily computed without any computational problem. Let $\tilde{\mathbf{f}}_i^t$ denote the sum of all spring forces and viscosity force exerted on the mass-point i, and $\mathbf{f}^t$ be the vector of the forces $[\tilde{\mathbf{f}}_1^t, \tilde{\mathbf{f}}_1^t, \cdots, \tilde{\mathbf{f}}_n^t]^{\mathbf{T}}$. Then the problem shown in Eq.3 is efficiently represented as follows:

$$\mathbf{W}\Delta\mathbf{v}^{t+h} = h\tilde{\mathbf{f}}^t \tag{4}$$

By taking into account the property of the matrix $\mathbf{W}$, the velocity change of each mass-point can be described as follows:

$$\Delta\mathbf{v}_i^{t+h} = \mathbf{W}_{ii}^{-1}(h\tilde{\mathbf{f}}_i^t + h^2 \sum_{(i,j)\in E} \mathbf{J}_{ij}\Delta\mathbf{v}_j^{t+h}) \tag{5}$$

The update formula for mass-point i shown in Eq.5 depends on the same formula for mass-point j. Therefore, we cannot directly update the velocity change of each mass-point. Our method computes the approximate solution with an iterative update based on Jacobi iteration as follows:

$$\begin{aligned} \Delta\mathbf{v}_i^{t+h^{(0)}} &= \mathbf{W}_{ii}^{-1}h\tilde{\mathbf{f}}_i^t \\ \Delta\mathbf{v}_i^{t+h^{(k+1)}} &= \mathbf{W}_{ii}^{-1}(h\tilde{\mathbf{f}}_i^t + h^2 \sum_{(i,j)\in E} \mathbf{J}_{ij}\Delta\mathbf{v}_j^{t+h^{(k)}}) \end{aligned} \tag{6}$$

where $\Delta\mathbf{v}_i^{(k)}$ denotes the velocity change computed with k iterations.

Fig.3 shows the result of the stable integration described in this paper. Since the mass-spring is sufficiently stiff, the model does not stretch or contract severely. However, the resulting animation does not look realistic because no damage is preserved on the surface.

3.2 Breakable Hinge Springs

Although the stable integration method described in the previous subsection enables real-time animation of a stiff mass-spring model, the resulting images or animations produced with the method are far from the real appearance and behavior of paper.

The unsatisfactory result is caused by the static structure of the mass-spring mesh. When real paper is deformed by external forces, the paper is easily damaged and the damage is preserved on the surface. However, the stiff mass-spring mesh cannot represent such characteristic.

In order to represent the damage preserving property of paper object, we propose a breakable hinge spring model. The breakable hinge spring model is an adaptive mass-spring mesh structure. In this model, each spring can be broken when its length is shorten by external or internal forces. When a spring is broken, the mesh structure of the paper model should also be modified in order to maintain the mass-spring mesh to be a triangular mesh.

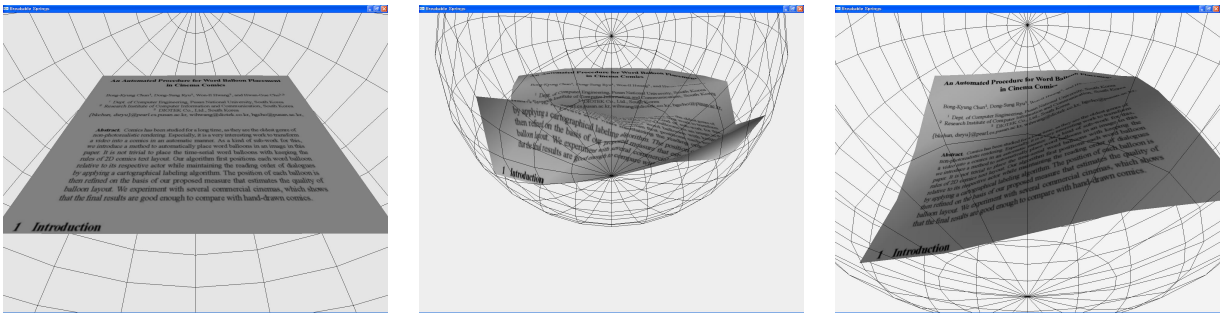


Fig. 3 Stable simulation of a stiff mass-spring model without breakable springs

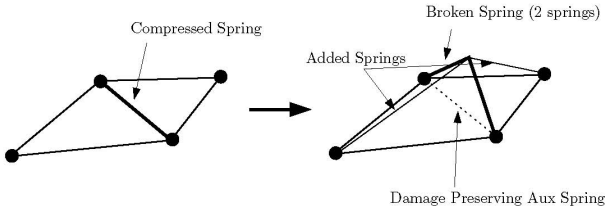


Fig. 4 Breakable spring and mesh adjustment

Fig.4 illustrates the breakable spring model. On the left side of the figure, a spring edge between two triangles is compressed. The compressed spring is broken and the adjusted mesh structure is shown on the right side. The compressed spring is divided into two distinct spring edges in order to maintain the original length of the compressed spring. The broken spring edge causes a problem in the mesh structure. Because of the broken edge, the mesh is not anymore a triangular mesh. In order to fix the problem, two additional springs are used. After adjusting the mesh structure, the breakable spring model inserts an auxiliary spring to preserve the damage.

In the Fig.4, the auxiliary spring is represented with a dotted line. Because of the auxiliary spring, the mesh remembers the damage and shows the paper-like wrinkles. Fig.5 shows the mesh appearance when springs are randomly chosen and broken. The thick blue lines are breakable springs, and thin red lines are auxiliary spring edges which are not breakable. As shown in the figure the breakable spring model can efficiently represent the damages on the paper surface.

Although the breakable spring model produces paper-like appearance, the method has a severe disadvantage: breakable springs can generate wrinkles only across the existing spring edges. Therefore, it is impossible to preserve the damage when the paper is folded along the existing springs. In order to represent the wrinkles along the spring edges, the springs also have to work like hinge joints. The hinge folding of a spring edge is implemented by applying an auxiliary spring across the spring edge as shown in Fig.6. In the figure $\mathbf{x}_i$ and $\mathbf{x}_j$ denote the

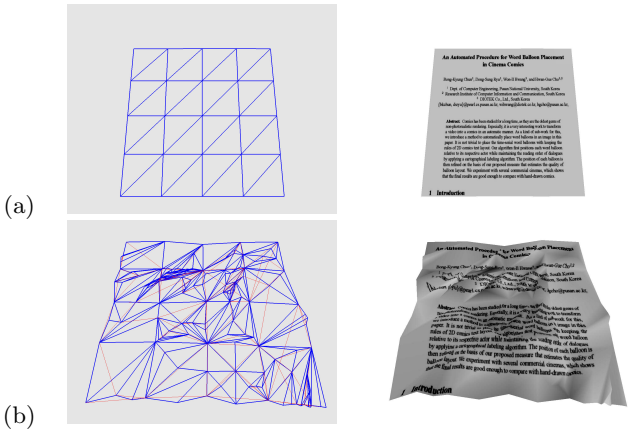


Fig. 5 Geometric appearance with breakable springs: (a) original mesh, (b) result when springs are randomly broken

mass-points linked with a spring, and $\mathbf{x}_r$ and $\mathbf{x}_l$ are the mass-points on the right and the left of the spring respectively. If the distance between $\mathbf{x}_r$ and $\mathbf{x}_l$ is shorter than a specified threshold, an auxiliary spring between the mass-points is inserted to the mass-spring structure.

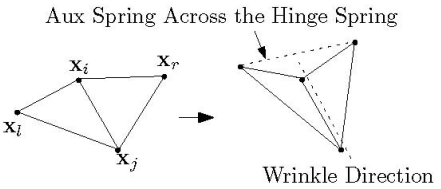
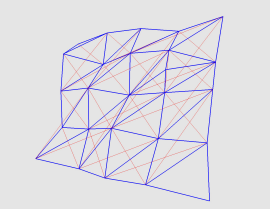


Fig. 6 Auxiliary spring across a hinge folding spring

Fig.6 shows the result when auxiliary springs for hinge springs have been inserted. It can be seen that the hinge springs make the surface of the deformable object wavy.

The method proposed in this paper utilizes both the breakable springs and hinge springs. Each spring in the mesh is either a breakable hinge spring or an unbreakable auxiliary spring. The result by randomly breaking or folding the spring edge is shown in Fig.8. As shown in



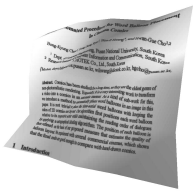


Fig. 7 Geometric appearance with hinge springs

the figure, breakable hinge springs make the mesh structure look like paper.

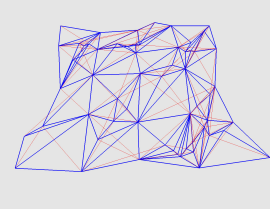


Fig. 8 Auxiliary spring across a hinge folding spring

3.3 Adaptive Mesh Structure

Although the mesh with randomly broken hinge springs displays plausible paper appearance, the random approach cannot be applied to paper animation. In the animation sequence, the paper model should be damaged by the forces.

In order to produce a plausible animation sequence, an automatic edge selection strategy for breaking or folding springs is necessary. Let $\mathbf{x}_i$ and $\mathbf{x}_j$ be the mass-points linked with a breakable spring. The locations of the mass-points at the original rest state are denoted as $\mathbf{x}_i^0$ and $\mathbf{x}_j^0$, and the locations at the current time t are $\mathbf{x}_i^t$ and $\mathbf{x}_j^t$. The rest length of the spring $l_{i,j}^0$ is, therefore, $|\mathbf{x}_i^0 - \mathbf{x}_j^0|$. The current length of the spring $l_{i,j}^t$ is, similarly, $|\mathbf{x}_i^t - \mathbf{x}_j^t|$. Because the spring is broken when it is compressed, the ratio $l_{i,j}^t / l_{i,j}^0$ is the most important value for determining whether the spring will be broken or not.

A simple approach is to break the spring when the ratio $l_{i,j}^t / l_{i,j}^0$ is less then a threshold. However, the damage on the paper surface actually occurs stochastically. Therefore, we used the ratio as the parameter for computing the probability of the catastrophic phenomena in the mesh structure. In other words, the ratio ranges [1,0] for a compressed spring, and the probability of breaking the spring $\mathbf{P}_{i,j}$ is computed as follows:

$$\mathbf{P}_{i,j} = (1 - l_{i,j}^t / l_{i,j}^0)^{\phi} \tag{7}$$

where ϕ is a parameter that controls the fragility of the springs.

When a spring is broken, a new mass-point must be inserted. The location of the mass-point must be determined. An easy method is to set the location of the new mass-point as the middle of the two mass-point linked with the original spring. However, this simple method recursively breaks the newly inserted broken springs. We adjust the location of the mass-point along the surface normal vector. The magnitude of the adjustment movement h can be easily computed as follows:

$$h = \frac{\psi}{2} \sqrt{(l_{i,j}^0)^2 - (l_{i,j}^t)^2} \tag{8}$$

where ψ is a parameter that controls the area conservation property of the paper model.

3.4 Fractal Surface Texture

The modeling of surface details of paper in computer graphics is a little explored subject. Several researchers have investigated the properties of paper as a canvas, i.e. its interaction with ink and different types of pens and paint. However, in general only surface properties with respect to absorption and diffusion of liquids are considered for these types of applications.

Baxter et al. model the advection of paint on paper and incorporate adhesion and absorption by clamping the flux such that at least some paint stays on the part of the surface covered by it [2]. Chu and Tai have developed an advanced brush model for Chinese calligraphy [3]. The interaction of the brush with the paper is described by a friction index which represents the resistance of the porous paper surface. The deposition of ink onto the paper surface is dependent onto the contact area and does not take into account physical properties of the paper. Rudolf et al. present a model of wax crayons [11]. The underlying paper model uses a height-field texture which was synthesized by convolving a quarter-circle arc with a lattice populated by uniform noise.

A more advanced paper model has been presented by [8]. The authors use a three layer model which is based on the observation that pigments and water of paint form a layer on top of the paper, where they act like a shallow body of water. After some time pigments are deposited onto the paper surface where they form a *pigment layer*. Finally the *capillary layer* represents the inner paper structure and is responsible for the movement of absorbed water. The authors model this layer by using a procedural texturing method [14].

3.4.1 Paper Surface as Fractal Terrain

When looking at real paper it can be observed that its difficult for humans to observe the object geometry from the shading changes itself. However, when texture mapping images of real paper onto a flat surface the result

looks unrealistic because of the straight object boundaries and the lack of height differences (straight silhouette) and lack of self-occlusion. In order to overcome these problems we use a fractal model and texture map it with a photo of real paper. The fractal model is constructed by displacing the corners of the paper by random amounts and bilinearly interpolating the surface in between. In each iteration each bilinearly interpolated patch is subdivided into four equally sized regions, the corners of which are again displaced by a random amount where the maximum possible displacement is at most half of the amount used in the previous iteration.

The process can be implemented more efficiently by using a loop rather than a recursion. Let d_{max} be the maximum displacement on the top level and let's assume for simplicity that the paper is modeled using $2^N + 1$ equally spaced vertices in each direction. Then in iteration $i = 0, \ldots, N$ the vertices $(k * 2^{N-i}, l * 2^{N-i})$ are displaced by $d_{max}/2^i$ where $k, l = 0, \ldots, 2^i$. The vertices within each patch

$$[(k2^{N-i}, l2^{N-i}), ((k+1)2^{N-i}, l2^{N-i}),$$
$$((k+1)2^{N-i}, (l+1)2^{N-i}), (k2^{N-i}, (l+1)2^{N-i})]$$

are bilinearly interpolated. Note that in the last step $(i = N)$ all vertices are displaced and hence no bilinear interpolation is necessary.

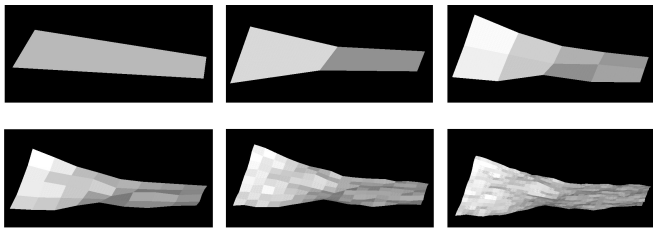


Fig. 9 Fractal model of crumbled paper rendered using flat shading to indicate bilinearly interpolated patches. The images show the surface after the first six iterations (top-left: after 1 iteration, bottom-right: after 6 iterations).

Figure 9 shows the results of this process after the first (top-left) and the first six iterations (bottom-right). This type of fractal model works well for terrains, but has several drawbacks when modeling paper.

The first drawback is that the resulting surface contains many high-frequency components. This leads to an excessive jagginess of the silhouettes and, when using shading, leads to a very unrealistic pattern of illumination changes. In order to prevent these artifacts we employ a law-pass filter. Efficiency is increased by using a separable 1D filter which is applied both parameters directions of the paper surface. The filter has the components

$$[0.006, 0.061, 0.242, 0.382, 0.242, 0.061, 0.006]$$

which sum up to one. The tensor product of these values gives a close approximation of a 7×7 Gaussian filter.

The second drawback is that the displacement of vertices in z-direction increases the surface area of the object. Since paper is a virtually incompressible material an increase in surface area is undesirable. Furthermore the outline of the displaced surface in figure 9 is still a rectangle which will look unrealistic when viewed from the top. We prevent these artifacts by iterating trough the grid cells of the surface in both coordinate direction and by shifting vertices such that the length of each grid cell is $l/2^N$ where l is the side length of the undeformed paper and $N + 1$ is the number if vertices along that side. Note that we first shift all vertices in x-direction and then in y-direction, which might change the grid length in x-direction. However, experiment showed that this effect was negligible and that this simple method generated suitable results.

3.4.2 Rendering and Texture Manipulation for Fractal Paper

Figure 10 (a) shows the paper model texture mapped with a photograph of crumbled paper. The model looks fairly realistic. Some simple experiment showed that without detailed investigation an observer is unlikely to notice that the model geometry does not correspond to the folds and crumbles from the paper texture. However, when moving the paper around the texture map's color stays unchanged whereas for a real piece of paper the viewer will see shading differences depending in the paper's orientation with respect to the scene's light sources.

In order to incorporate shading we render the surface in white, illuminate it and blend the color with the texture map using the OpenGL blending facilities. The result is shown in figure 10 (b). The result does not look convincing because the fractal model has many small bumps which do not correspond with the geometry of the paper used in the texture map. We found that better results are obtained by using the surface normals of an intermediate (lower resolution) step of the fractal surface, which does not contain any high frequency components. Figure 10 (c) and (d) show the results obtained by using the surface normals of the fractal surface after two and four iterations, respectively.

We can apply our model to textured objects by using the multi-texturing extension of the OpenGL standard which is available for most modern graphics cards. Figure 11 shows several examples which were rendered using our fractal model for the geometry and a paper texture to give fine shading details. Part (a) of the figure shows a cartoon image. Note that the radial lines of the image are bend according to the fractal geometry which is different from the actual geometry corresponding to the paper texture. However, we believe that without prior knowledge this difference is not noticeable. Part (b) of the figure shows a bank note which is an interesting example since it has extremely fine texture details and users are very familiar with its appearance. Part (c) shows a

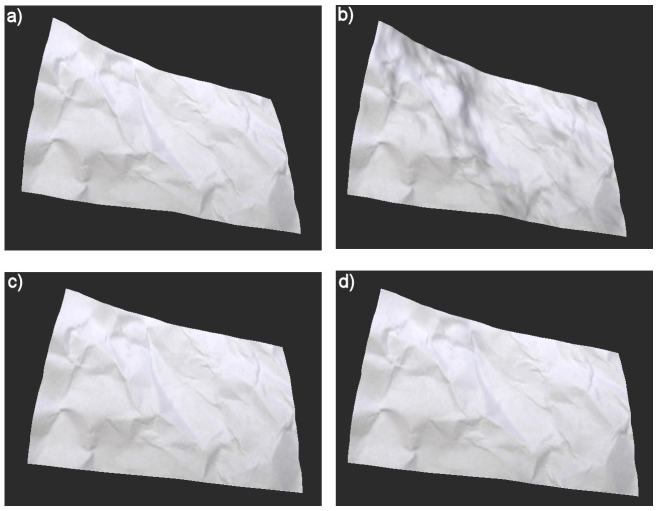


Fig. 10 The fractal model texture mapped with a photograph of crumbled paper (a), blended with the illuminated fractal surface, and blended with an illuminated low-resolution surface after two (c) and after four (d) iterations.

journal cover with a low level of wrinkled surface texture and fractal deformation. Finally part (d) shows another bank note model with low level of surface perturbations and wrinkled surface texture and folded.

More users studies are necessary to better understand the perceived realism of our fractal model under different circumstances.

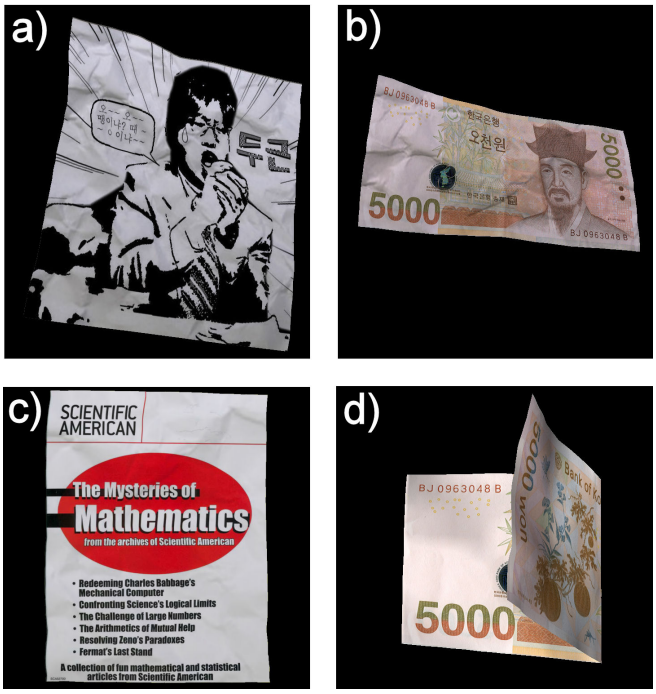
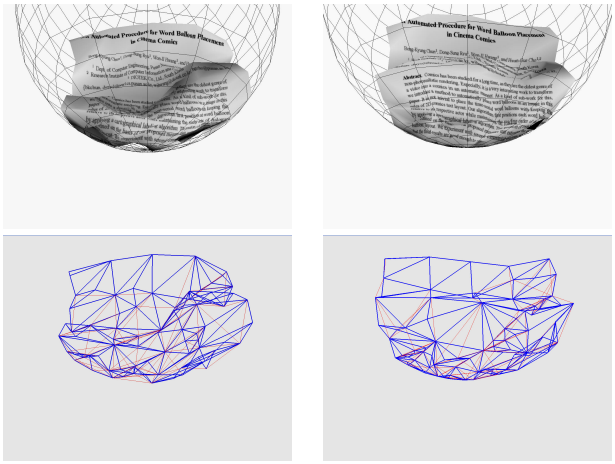


Fig. 11 The fractal model and paper texture used to render a cartoon image (a) and a bank note (b). [Cartoon image used with kind permission of Mr. Dong-Sung Ryu]

4 Experiment and Results

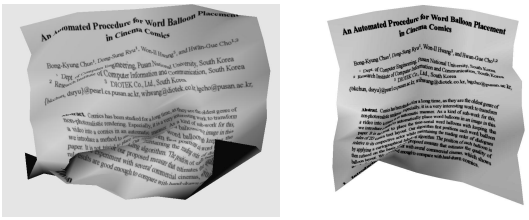
The control parameters for the proposed method are ϕ and ψ in Eq.7 and Eq.8 respectively. According to the control parameters, the appearance and the behavior of the paper model is changed.



(a)ψ=1.0 (b) ψ=0.1

Fig. 12 Effect of control parameter ψ

Fig.12 shows the effect of the control parameter ψ. As shown in the figure, a large ψ value generates stiff paper such as card board which you can not bend and which folds if you apply to much force. And a small ψ value generates flexible paper which does not crumble easy (e.g. a bank note).



(a) ϕ=1 (b) ϕ=5

Fig. 13 Effect of control parameter ϕ

Fig.13 shows the effect of the control parameter ϕ. The parameter controls the fragility of the paper. Therefore, a large ϕ value produces rubbery paper while a small ϕ produces fragile paper.

Fig.14 shows the animation result when breakable hinge springs are employed in an interactive environments. The springs are broken or folded according to the external force exerted by the sphere. As shown in the figure, the breakable hinge spring model can produce plausible paper animation.

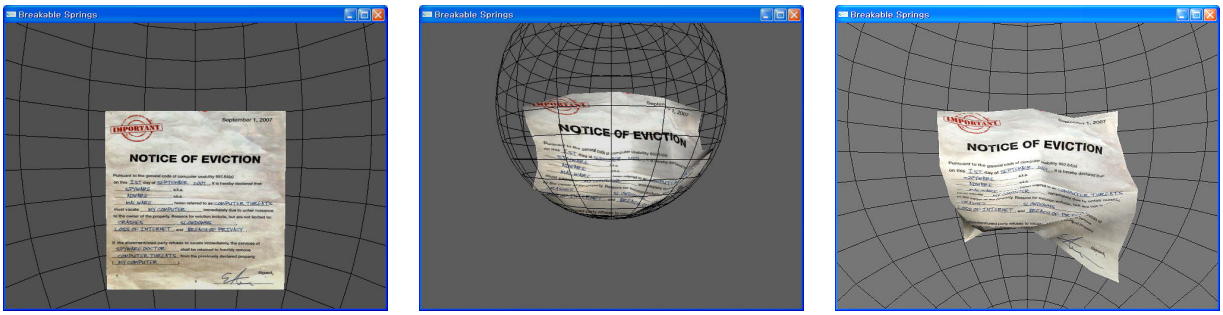


Fig. 14 Interactive animation of the paper model with breakable hinge springs

5 Conclusion

In this paper, we proposed a simple and intuitive approaches to animate the damage-preserving stiff mass-spring model in order to represent the behavior of paper in virtual environments. The proposed method exploits an approximate implicit method to guarantee the stability of the system, and employs breakable hinge springs to show damage-preserving properties. A fractal model is used to create fine surface details.

The proposed method enables real-time animation of paper objects in virtual environments or game application. The proposed method is intuitive because it is based on a simple and effective mass-spring model and uses a direct update formula instead of solving the large linear system.

The proposed method produces stable paper animations and generates natural-looking wrinkles on the paper surface. Besides the realistic appearance, the proposed method also produces paper-like moving behavior in interactive environments, and can be easily implemented. The proposed method can be successfully utilized for paper animation in realtime or interactive applications.

References

1. Baraff, D., Witkin, A.: Large steps in cloth simulation. Proceedings of SIGGRAPH 98 pp. 43–54 (1998)
2. Baxter, W., Wendt, J., Lin, M.C.: IMPaSTo: a realistic, interactive model for paint. In: NPAR '04: Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering, pp. 45–148. ACM Press (2004)
3. Chu, N.S.H., Tai, C.L.: Real-time painting with an expressive virtual chinese brush. IEEE Computer Graphics and Applications **24**(5), 76–85 (2004)
4. Chu, Y.C., Witten, I.H., Lobb, R., Bainbridge, D.: How to turn the page. In: Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries (JCDL '03), pp. 186–188. IEEE Computer Society Press (2003)
5. Desbrun, M., Schröder, P., Barr, A.: Interactive animation of structured deformable objects. Graphics Interface '99 pp. 1–8 (1999)
6. Ida, T., Buchberger: Proving and solving in computational origami. In: 5th International Workshop for Symbolic and Numeric Algorithms for Scientific Computation, pp. 246–263 (2003)
7. Kang, Y.M., Cho, H.G.: Real-time animation of complex virtual cloth with physical plausibility and numerical stability. Presence - Teleoperators and Virtual Environments **13**(6), 668–680 (2004)
8. Laerhoven, T.V., Reeth, F.V.: Real-time simulation of watery paint. Journal of Computer Animation and Virtual Worlds **16**(3), 429–439 (2005). URL: http://research.edm.luc.ac.be/~vanlaerhoven/publications/vanlaerhoven_vanreeth_casa2005.pdf [Last accessed 18/12/2007]
9. Meyer, M., Debunne, G., Desbrun, M., Bar, A.H.: Interactive animation of cloth-like objects in virtual reality. The Journal of Visulalization and Computer Animation **12**, 1–12 (2001)
10. Ng, H.N., Grimsdale, R.L.: Computer graphics techniques for modeling cloth. IEEE Computer Graphics & Applications **16**(5), 28–41 (1996)
11. Rudolf, D., Mould, D., Neufeld, E.: A bidirectional deposition model of wax crayons. Computer Graphics Forum **24**(1), 27–40 (2005)
12. Terzopoulos, D., Platt, J., Barr, A., Fleischer, K.: Elastically deformable models. Computer Graphics (Proceedings of SIGGRAPH 87) **21**(4), 205–214 (1987)
13. Volino, P., Courshesnes, M., Magnenat-Thalmann, N.: Versatile and efficient techniques for simulating cloth and other deformable objects. Proceedings of SIGGRAPH 95 pp. 137–144 (1995)
14. Worley, S.: A cellular texture basis function. In: SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pp. 291–294. ACM (1996)