# Paper Fracture Simulation Based on FEM

Takao Sato, Makoto Fujisawa and Kenjiro T. Miura
Graduate School of Science and Technology
Shizuoka University
Address 3-5-1 Johoku, Hamamatsu, Shizuoka, 432-8561, Japan

## Abstract

Recently researches on techniques to use simulations based on the physical laws for CG animations are becoming popular and popular. Although paper exists ubiquitously around us in daily life, it is not straightforward to express its folding lines, fractures, and textures and not much effort has been made on researches of paper for CG. Hence the purpose of this paper is to simulate paper fracture based on the physical laws and use the results for CG animations as the first step to the CG-oriented paper simulation.

The fracture simulation of paper is performed at first by modeling a piece of paper with triangular meshes and applying the FEM to calculate their stresses and strains. Then we determine the starting position of the fracture and the direction of the crack and rebuild the structure of the meshes to express the fracture cross sections. Furthermore for the cross section originally expressed by piece-wise linear segments, by use of the turbulence function we generate natural-looking geometrically complicated cross sections as textures and map them to the meshes to represent the paper characteristics that become clear when the piece of paper is torn apart. The geometry of the fracture cross sections remains similar even though the mesh size is changed because of the noise-mixed texture mapping, that allows us to use relatively coarse meshes and simulate the paper fracture in almost real time.

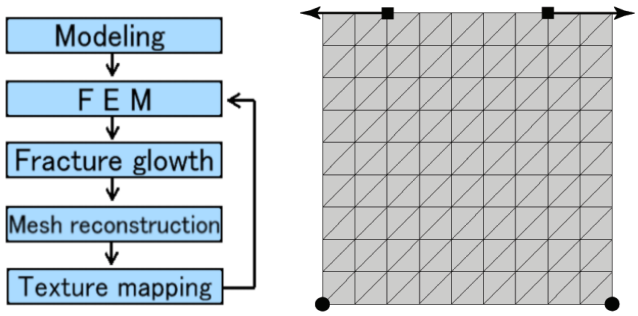
**Keywords:** fracture, paper, FEM, texture mapping, turbulence function

## 1 Introduction

The physical laws are used to simulate natural phenomena to produce CG animations in a lot of researches. For example, the behaviors of water flows, flame and human clothes have been realistically reproduced as CG animations and they have been used for movies and games. Although paper exists ubiquitously around us in daily life as water and clothes, not much effort has been made on researches of paper for CG. The cloth simulation is one of the researches on two dimensional objects like paper and it has been well studied. However, it is not straightforward to express its folding lines, fractures, and textures. Hence the purpose of this paper is to simulate paper fracture based on the physical laws and use the results for CG animations as the first step to the CG-oriented paper simulation.

## 2 Simulation Outline

Figure 1(a) shows the algorithm of the paper simulation in this research. At first, a piece of paper is




(a) Algorithm (b) Simulation conditions

Figure 1: Simulation outline

modeled by a triangular mesh and the Finite Element Method (FEM) is applied to the model. The stiffness equation is used as a motion equation and the stresses and displacements of the nodes of the model are calculated. Next, the fracture algorithm determines a fracture position and direction by use of the node stresses and displacements obtained by the FEM. Then the mesh is reconstructed to appropriately express a clack generated by the fracture. Finally, the paper texture is represented by mapping a suitable image to the model. The processes from the clack generation to the texture mapping are grouped as one step and the fracture is simulated by applying this step repeatedly.

In this paper, all simulation results are performed under the same conditions as shown in Fig.1(b). We apply two forces of the same magnitude in the opposite directions at the nodes illustrated as rectangular marks, fix the two nodes displayed by circular marks and make other nodes free.

# 3 FEM Principle

Each process mentioned in the previous section is explained in detail.

## 3.1 FEM

A piece of paper is modeled by a triangular meshe with nodes and triangular elements as shown in Fig.2. The FEM is used to calculate the node stresses and displacements in this research.

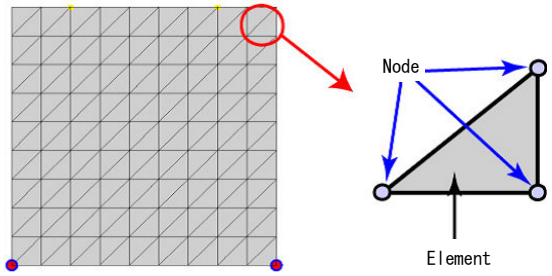



Figure 2: Paper model

### 3.1.1 Motion equation

The motion equation used for the FEM is a stiffness equation given by

$$\{\mathbf{F}\} = [K]\{\mathbf{u}\} \tag{1}$$

where $\mathbf{F}$ is loads, $K$ is spring constants, and $\mathbf{u}$ is displacements. One side of each triangular element of the model is regarded as a truss element. The spring constant of each truss element is determined by the following equation:

$$k = \frac{EA}{l} \tag{2}$$

where $E$ is its Young's modulas, $A$ is its cross-section area, and $l$ is the distance of its two nodes. At first, the stiffness matrix of the truss element for each side of the triangle is obtained and the stiffness matrix of the triangular element is calculated. The total stiffness matrix of the whole mesh is constructed by superpositioning the stiffness matrix of each triangular element. Using the whole stiffness matrix and the load-displacement vector, the stress of each node is calculated. Then the displacement of each node is determined. The piece of paper is deformed by repositioning the node position according to the calculated displacements.

## 3.2 Fracture growth

The FEM can calculate the node displacements and the stresses of the elements, but the piece of paper is only deformed elastically and not broken. Hence it is necessary to determine the breaking position and direction of the paper based on the stresses obtained by the FEM


Figure 3: Truss element

### 3.2.1 Fracture type

It is possible to consider two types of the fracture for the simulation. One is the fractures started from edge elements and the other is those started from nodes.

In case of the fractures started from edge elements, along the straight line shown in Fig.4(a) as a dashed line determined by the stress distribution, a clack should be generated through the center of gravity of the triangular element as shown in Fig.4(b). It causes a problem how to grow the clack in the next step. Another problem is that it generates nodes and triangular elements more than the fractures started from nodes. Therefore we adopt the fractures started from nodes because of the low cost of processing time and the ease of the implementation of the mesh reconstruction.

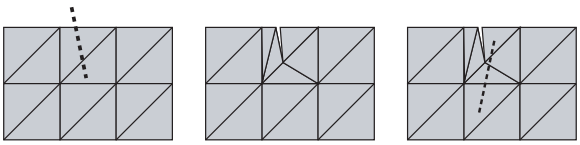



Figure 4: Fracture started from an edge

The FEM calculates the stresses applied to the triangular elements. Hence it is necessary to transform the stresses of the triangular elements to the node stresses to obtain the stresses applied to the nodes. We apply the method proposed by O'Brien et al. for 3D objects[2] to our 2D problem to calculate the node stresses.

### 3.2.2 Stress decomposition

The stress applied to a triangular element is decomposed to tension and compression components. The tension component $\sigma^+$ and the compression component $\sigma^-$ of the element stress are given by

$$\sigma^+ = \sum_{i=0}^{2} \max(0, v^i(\sigma)) m(\hat{\mathbf{n}}^i(\sigma)) \tag{3}$$

$$\sigma^- = \sum_{i=0}^{2} \min(0, v^i(\sigma)) m(\hat{\mathbf{n}}^i(\sigma)) \tag{4}$$

respectuvely where $v^i(\sigma)$ is the $i$-th eigenvalue of $\sigma$ and $\mathbf{n}^i(\sigma)$ is its eigenvector. $m(\mathbf{a})$ is a $3 \times 3$ symmetrical matrix whose eigenvalues are given by $|\mathbf{a}|$ where $\mathbf{a} \in \Re^3$.

$$m(\mathbf{a}) = \begin{cases} \mathbf{a}\mathbf{a}^T/|\mathbf{a}| & \mathbf{a} \neq 0 \\ 0 & \mathbf{a} = 0 \end{cases} \quad (5)$$

The above equation decomposes the element stress to each components.

#### 3.2.3 Tensile and compressive forces at the nodes

We calculate the tensile and compressive forces at the nodes enforced by the triangular elements. The tensile force is calculated by

$$\mathbf{f}^+_{[\mathbf{i}]} = -\frac{s}{2} \sum_{j=1}^{3} \mathbf{p}_{[\mathbf{j}]} \sum_{k=1}^{2} \sum_{l=1}^{2} \beta_{jl} \beta_{ik} \sigma^+_{kl} \quad (6)$$

where $s$ is the area of the element and $\beta$ is obtained by the following equation from the world coordinates of the nodes $\mathbf{p}_{[\mathbf{i}]}$ shared by the triangular element:

$$\beta = \begin{bmatrix} \mathbf{p}_{[\mathbf{1}]} & \mathbf{p}_{[\mathbf{2}]} & \mathbf{p}_{[\mathbf{3}]} \\ 1 & 1 & 1 \end{bmatrix}^{-1} \quad (7)$$

Similarly $\mathbf{f}^-_{[\mathbf{i}]}$ is calculated for the compressive component. Both of the tensile and compressive components can be determined.

#### 3.2.4 Separation tensor

The separation tensor is calculated to determine the direction of the fracture. The separation tensor $\zeta$ is obtained from the tensile and compressive stresses acting on each node

$$\zeta = \frac{1}{2} \Bigg( - m(\mathbf{f}^+) + \sum_{\mathbf{f} \in \{\mathbf{f}^+\}} m(\mathbf{f}) + m(\mathbf{f}^-) - \sum_{\mathbf{f} \in \{\mathbf{f}^-\}} m(\mathbf{f}) \Bigg) \quad (8)$$

If the maximum positive eigenvalue of $\zeta$ of a node is larger than a certain threshold $\tau$, the material is broken at the node. In case that more than one eigenvalues exceed $\tau$, the material is broken at the node whose eigenvalue is the largest among them. The direction of the fracture in the world coordinate system is perpendicular to the eigenvector corresponding to the eigenvalue. The lengthes and directions of the arrows in Fig.5 indicate the magnitudes of the eigenvalues and the directions of the eigenvectors.

In Fig.5 the left and right loads are of the same strength in the opposite directions, but the fracture distribution is not bilaterally-symmetric. The cause of this asymmetry is considered to be the asymmetry of the model itself and related to the way of the mesh subdivision.

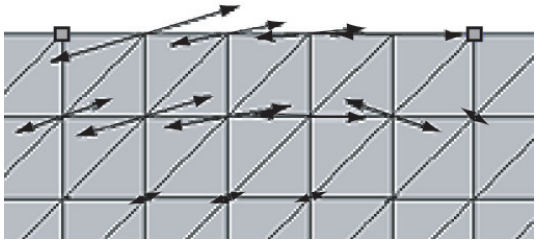


Figure 5: Eigenvalue and eigenvector

### 3.3 Mesh reconstruction

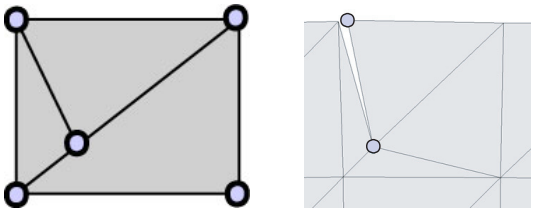
In the previous subsetion, we can determine the node where the fracture starts and the direction in which the fracture grows. The mesh is reconstructed to generate a clack at the node slanted in the breaking direction. The fracture direction does not always conform to the direction of one of the edges of the mesh and it is usually necessary to reconstruct the mesh along the fracture direction.

#### 3.3.1 Reconstruction method

The reconstruction is performed by duplicating the breaking node, generating a new node at the intersection of the line lying in the fracture direction and the boundary of the triangular element, and connecting the new node with the node generated from the broken node (circled points in Fig.6(b)).

#### 3.3.2 Important reminder

For the mesh reconstruction, if the node is located on one of the sides of the triangular element, the adjacent element becomes rectangular and it is not possible to model it with triangular elements (see Fig.6(a)). Hence to avoid such situations, the triangular element adjacent to the broken element is remeshed.

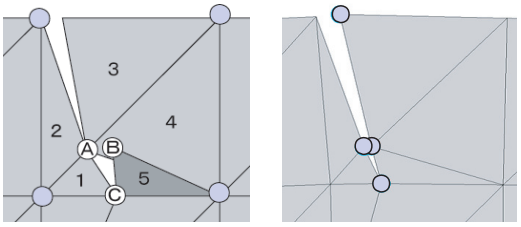


(a) Situation to be avoided (b) Reconstruction

Figure 6: Reconstruction in the first step

Although the reconstruction process so far is good enough for the first step of the fracture growth, the number of the elements to be reconstructed for the second and later steps increases. Although in the first step, the reconstruction is good enough to rebuild the triangular elements only directly related to the fracture as shown in Fig.6(b). In the second or later steps, it is necessary to reconstruct the non-broken elements

around the fracture. If the same treatment as the first step is applied to the mesh for the second or later steps, an incosistent mesh is generated where the elements are overlapped as shown in Fig.7(a). In this figure the node where the the fracture has started is indicated by A and at first new nodes indicated by B and C are generated by the reconstruction and the relationships among element 1, 5 and the node C are updated. In the second or later steps, element 3 and 4 are disconnected from the node A and then they are connected with the node B. This process can reconstruct the mesh correctly as shown in Fig.7(b). Note that the update of the mesh topology is not always applied to only two triangular elements. The number of the triangular elements connected with the node where the fracture is growing is not always equal to 4 although that is 4 in Fig6(b). Hence we have to consider the number of elements connected with the starting node to update the mesh topology. The toplogy update is carried out by these two processes.




(a) Inappropriate mesh (b) Reconstruction

Figure 7: Reconstruction in the second or later steps

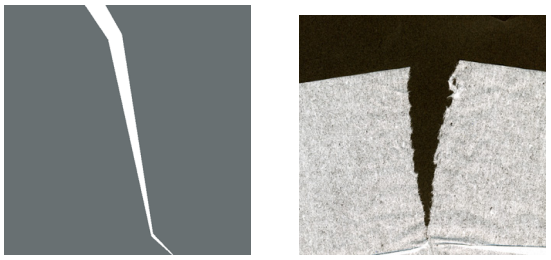
We have explained the whole process of the calculation of the fracture and the related reconstruction of the model so far. Then we recalculate the stiffness matrix and repeat the process mentioned above to grow the fracture.

## 3.4 Texture mapping

The fracture cross section of the paper after the mesh reconstruction is represented by a line segment since the section is an edge of the triangular mesh as shown in Fig.8(a). However the cross section of the real paper has ravels of fibers and it rarely becomes a straight line (see Fig.8(b)). Therefore we use a texture to repsent the cross sections.

### 3.4.1 How to treat texture

We represent the fracture cross sections by mapping textures generated by the noise function along them. We make two cross section lines facing each other a pair and adopt the same noise function to them to match the shapes of the two cross section lines. Although we use the random noise function, we can avoid overlaps of the cross sections and obtain natural looking cross sections by this technique.

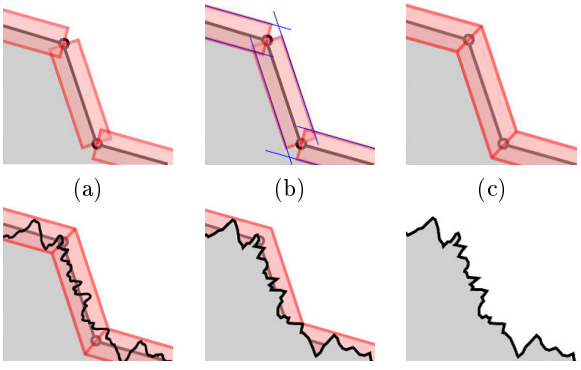


(a)Simulated cross section (b) Real paper

Figure 8: Comparison of fracture cross sections

### 3.4.2 Mapping texture along cross sections

At first we map a rectangular texture onto each edge of the cross sections. If we allocate rectangular textures along the cross seciton, it causes a problem of overlapping of the adjacent textures as Figure 9(a) illastrates the situation. Hence we avoid the overlapping by modifying the shapes of the textures to share the same edge between the two adjacent tetxures. The modification of the textures is carried out by calculating the intersections of the sides of the two consecutive textures parallel to the edge of the cross section (Fig.9(b)) and reshaping the textures using the intersection points as shown in Fig.9(c). Then we paint the texture inside in the paper color (Fig.9(d)) and outside in the backgroud color (Fig.9(e)) according to the noise function. We repeat the same process for the whole cross sections to map the textures.



(a) (b) (c)

(d) (e) (f)

Figure 9: Texture-mapping processes

### 3.4.3 Noise function

In this subsection, we explain about the noise function used in this research. There are several types of noise function including Perlin's noise and the white noise and Figure 10 shows the results of the applications of these noise functions to the texture mapping for the cross sections.

As shown in Fig.10 Perlin's noise has a clear periodicity and the white noise can not completely erase the linear shape of the cross section on the other hand. Both

(a) Perlin's noise (b) White noise

Figure 10: Noises

of them are not suitable to our application. Therefore we adopt the turbulence function[3, 4] to make the textures.

#### 3.4.4 Turbulence function

The turbulence function is given by

$$n_t(\mathbf{x}) = \sum_i^n \text{abs}\left(\frac{\text{noise}(2^i\mathbf{x})}{2^i}\right) \quad (9)$$

where noise($\mathbf{x}$) is a Perlin's noise at $\mathbf{x}$. The turbulance function is a superposition of of $n$ Perlin's noises of different periods. Figure 11 shows the variations of the textures with respect to the changes of $n$.

(b) n = 1 (c) n = 3 (d) n = 5

Figure 11: With turbulence functions of various $n$

We can obtain more natural looking cross sections by the turbulence function than the Perlin's or white noise function since we do not notice very much either the periodicity or the linearity of the cross sections. For a large $n$, the noise function with a large index $i$ close to $n$ in Eq.9 does not contribute the value of the turbulence function so much. Hence some $n$ is large enough depending on the image resolution of the simulation environment since the appearance of the textures remains almost the same for the larger $n$ than this value. For our experiments explained in the next section we used 5 for $n$ any value larger than which did not give us a better appearance of the texture.

The shape generated by the turbulence function is a quasi-fractral. A lot of natual objects have fractal nature and we can say it is more appropriate to use the turbulence function than other general noise functions without fractal nature to represent the cross section of paper because the piece of paper is made from natural materials.

| # of elements | # of nodes | Time (per one step) |
|---|---|---|
| 64 | 98 | 0.7 sec |
| 144 | 242 | 1.8 sec |
| 256 | 450 | 4.9 sec |
| 400 | 722 | 12.2 sec |
| 576 | 1058 | 22.2 sec |

Table 1: Processing time

## 4 Results

### 4.1 Simulation results

We can calculate the node stresses, simulate the fracture growth and achieve natural looking fracture cross sections generated by the fracture growth by use of the noise functions as shown in Fig.12 and 13.

### 4.2 Processing time

To incease the accuracy of the simulation, it is necessary to subdivide the mesh finer and make more triangular elements. The finer mesh can solve the problem of straight line cross sections that gives unnatural immpressions and we can obtain natural looking cross sections without texture mapping because of the high accuracy of the simulation. However it increses the number of the nodes and consequently the cost of the processing time per one step. Figure 12 shows the results of the 2-step interval until 10 steps of a 98-node mesh and Figure 13(b), (d), and (e) show those of a 1058-node mesh which has about 10 times nodes of the model of Fig.12 to increase the accuracy of the simulation. The results after several steps are shown which have similar clack widths to those of Fig.12. Table 1 shows the relationship between the numbers of the elements and the processing times per one step.

Without texture mapping, the results of the finer mesh is better than those of the coarser mesh since natual cross sections are generated without the lineality of the edges. With texture mapping, both of the results are similar although the growth of the fracture are somewhat different. However their processing times are very different to get the similar results with texture mapping. These results indicate that the coarse mesh gives almost the same outputs as the finer mesh in almost real time other than the accuracy of the simulation.

We can see some difference of the whole behavior of the deformation of the piece of paper although they have similar clack widths. One of the cause of the results is that we use the number of steps instead of time for the control of the simulation process. The values used for the deformation of paper are the solution of the stiffness equation obtained after loading forces to the model. The obtained displacements of the nodes take into account the plastic deformation since we do

not take care of the elastic deformation. The fracture caused by the plastic deformation depends on the mesh shape. It can be considered that the differences between the results of the meshes of different fineness derives from the difference of the number of loading and that of the amount of the plastic deformation.

# 5 Conclusions

We can represent the fray of paper at the cross section, one of the main feature of paper independently from the coarseness of the mesh in this research. Furthermore if the high accuracy of the fracture behavior is not required, we can perform fracture simulations of paper which yield natural looking cross sections similar to those with high accuracy in almost real time.

As future work, the irregularity of fiber orientations should be taken into consideration since the triangular mesh is regularly generated, the meterial properties of each node are isotropic and any anisotropic treatment has not been taken. To achieve more natural looking fracture simulations, it is important how to model anisotropy of paper. As mentioned in the previous section, the problem of the deformation difference bewteen the results of the meshes of different fineness caused by calculation accuracy should be solved. The typical fracture pattern of paper encountered in daily life is that by the three dimensional shear force, not that by the two dimensional tensile force. We will study how to implement it as future work.

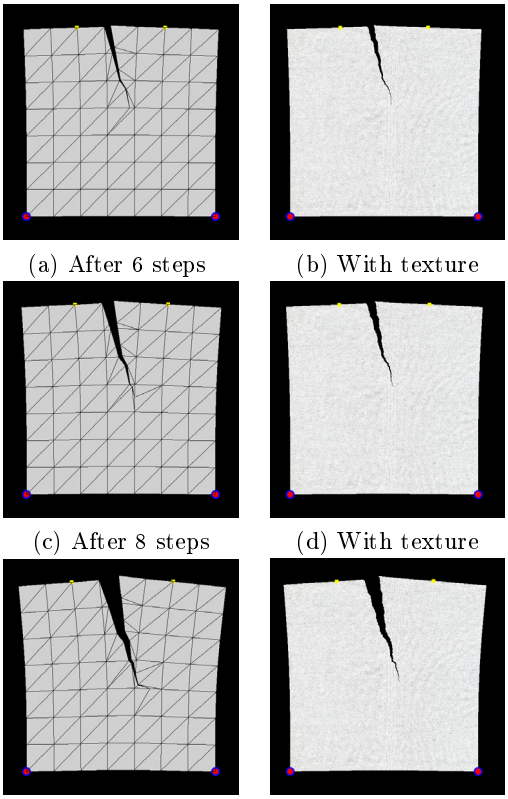
# References


[1] Nelson S.-H. Chu et al, "Real-Time Ink Dispersion in Absorbent Paper," SIGGRAPH2005, 2005.

[2] James F. O'Brien et al, "Graphical Modeling and Animation of Brittle Fracture," SIGGRAPH99, 1999.

[3] Ken Perlin, "An Image Synthesizer," SIGGRAPH85, 19(3), 1985.

[4] Ken Perlin and Eric M. Hoffert, "Hypertexture," SIGGRAPH89, 23, 1989.

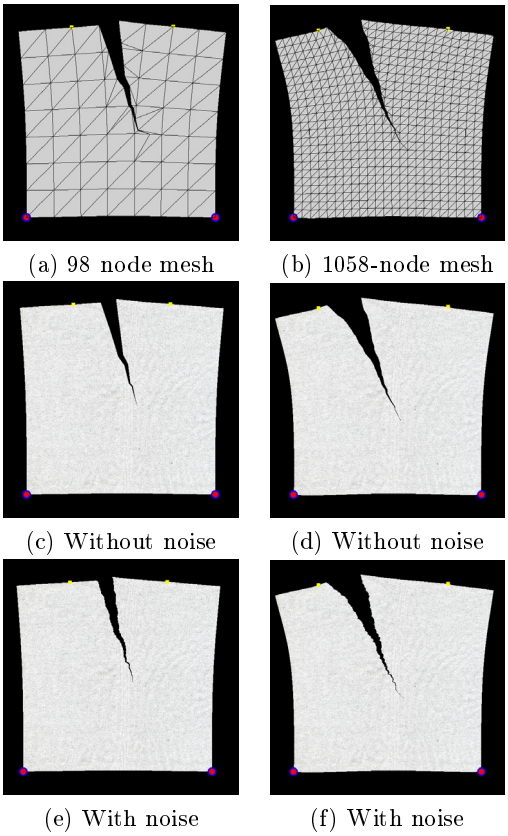



(a) After 6 steps (b) With texture

(c) After 8 steps (d) With texture

(e) After 10 steps (f) With texture

Figure 12: Results of 98-node mesh



(a) 98 node mesh (b) 1058-node mesh

(c) Without noise (d) Without noise

(e) With noise (f) With noise

Figure 13: 98- and 1058-node meshes