

Novel adaptive SPH with geometric subdivision for brittle fracture animation of anisotropic materials

Chen Li¹ · ChangBo Wang¹ · Hong Qin²

Published online: 6 May 2015
© Springer-Verlag Berlin Heidelberg 2015

Abstract In this paper, we articulate a novel particle-centric method to simulate the dynamics of brittle fracture for anisotropic materials. The key motivation of this paper is to develop a new hybrid, particle-based simulation that inherits advantages from both powerful finite element methods and popular mesh-free methods, while overcoming certain disadvantages of both types of methods. Our method stems from two novel aspects: (1) a physical model built upon an improved mechanical framework and the adaptive smoothed particle hydrodynamics (SPH), an improved variant of traditional SPH, which can handle complicated anisotropic elastic behaviors with little extra cost; and (2) a hybrid, adaptive particle system that serves for more accurate fracture modeling with richer details. At the physical level, in order to facilitate better control during the formation of fracture and improve its time performance, we develop a physical framework based on contact mechanics and adopt the stress and energy analysis on the anisotropic SPH numerical integration to pinpoint fracture generation and propagation. At the geometric level, in order to reduce time consumption and enhance accuracy in rigid dynamics and fracture generation, we employ hybrid, fully adaptive particles in the vicinity of fracture regions via geometric subdivision. Our novel approach can facilitate the user to control the generation of cracks with low computational cost and retain high-fidelity crack details during animation. Our comprehensive experiments demonstrate the controllability, effectiveness, and accuracy of our

method when simulating various brittle fracture patterns for anisotropic materials.

Keywords Brittle fracture · Anisotropic material · Adaptive tetrahedral subdivision · Animation control

1 Introduction

Brittle fractures such as shattering glass or broken porcelain are commonly seen phenomena in movie special effects and digital video games. Physically correct animation of detailed fracture still remains a difficult problem in computer graphics in spite of many published literatures on this subject. When processing the fracture due to the numerical complexity and instability in mechanical models, it tends to suffer from expensive computational costs. Other challenges result from the less-accurate generation and propagation of cracks that are also far less user-controllable. Alternatively, the non-physically based pre-fracture models are popular for the simplicity purpose in game and video industry, but the animation result generally falls short to match with the reality demand, and also manual design is time-consuming. So it is necessary to ease the burden on computational costs in rigid dynamic simulation while obtaining high-quality visual results in physically correct fracture animation.

Recent years have been witnessing some progresses for brittle fracture in computer graphics. Finite element method (FEM) [1,2] is used extensively in physically based deformation for simulating fracture; however, the FEM computation is sometimes costly to maintain high-quality meshes. On the other hand, meshless methods are popular thanks to their ease of implementation and interactive applications. The unstructured scheme makes it suitable for simulating topology-changing phenomena such as fracture. It may

✉ ChangBo Wang
cbwangcg@gmail.com

¹ Software Engineering Institute,
East China Normal University, Shanghai, China

² Department of Computer Science, Stony Brook University,
Stony Brook, USA

be noted that deformable solid simulation during the past decade [3,4] could be computationally enabled with either expensive and complex implicit methods guaranteeing stability or less-robust explicit methods.

For brittle fracture and its numerical simulation, anisotropic materials are much more difficult to handle and process due to a large variety of fracture patterns existed with different distributions of anisotropic materials, frequently more than twenty parameters in anisotropic elasticity are called in, making it a challenge to achieve a fast and controllable simulation. Even worse, anisotropic mesh generation and analysis are more complex and time-consuming. The adaptive remeshing technique has demonstrated its efficiency and effectiveness recently, Narain et al. [5] introduced anisotropic meshes in cloth simulation, but their approach has limitations when dealing with incompressible materials and self-collision. In general, anisotropic particle methods suffer from heavy computational expenses when searching neighbors and maintaining stability.

In this work, we propose a novel particle-based approach to simulate brittle fracture of anisotropic materials. A simple and stable representation is initialized with Delaunay-driven tetrahedral generation based on FEM domain decomposition. Considering the structure of anisotropic materials, the weighted Principal Component Analysis (WPCA) is applied to adjust anisotropic kernels. Our physical framework adapts the linear elasticity model in contact mechanics and adaptive Smoothed Particle Hydrodynamics (SPH) which uses dynamic anisotropic kernels to analyze the strain tensor in order to capture deformation in different directions with minimum computational costs. In this way, the simulation is stable, controllable, and convenient for interactive applications. To obtain richer cracking details, we apply a geometric crack scheme based on anisotropic physical analysis to contact area, then proceed to the next step by introducing a fully adaptive particle model with the self-refinement ability, which could improve accuracy for fracture track and avoid artifacts. Equipped with geometric refinement, a coarse particle representation could improve the overall computational efficiency during collision query. The main contributions of this paper are:

- *A new physical framework of brittle fracture based on contact mechanics and adaptive SPH with anisotropic kernels.* Since our method employs the classic theory of linear elasticity to calculate the displacement field and introduces a novel numerical solver to control the generation of cracks based on adaptive SPH approximation of displacement field, the amount and direction of the cracks can be managed via the specified setting of parameter combination. The complex stress model for anisotropic materials could be greatly simplified.

- *An user-friendly and controllable scheme for cracking of anisotropic materials over contact areas.* We offer a simple yet efficient way for the user to achieve desirable fracture effects on rigid objects. Except that Young's modulus and Poisson ratio are employed to define the hardness of rigid, anisotropic tensor could influence fracture effects along various directions by way of user-specific local geometry analysis. A crack generation scheme based on energy analysis over contact areas is devised to warrant visual realism.
- *An efficient adaptive particle model for simulating the fracture and crack of rigid objects.* We propose an efficient adaptive particle model to approximate rigid objects. The model adopts a small number of physical particles to initialize the rigid object according to the Delaunay mesh in the same way as FEM does, and performs strain and stress analysis on the original model. Adaptively subdivided particles of high resolution can be dynamically regenerated in the neighborhood of contact and crack areas for accurate fracture tracking, while coarse boundary particles could accelerate collision detection during simulation. The efficiency of our framework makes it possible for modeling anticipated object details and accurate subsequent motion.

The rest of this paper is organized as follows. After briefly reviewing the previous works in Sect. 2, the linear elasticity model for stress analysis is introduced in Sect. 3. In Sect. 4, the anisotropic SPH solver and the analysis scheme based on it are elaborated. Then we describe the adaptively subdivided model for tracking fracture and crack, and detail our schemes in handling geometric subdivision and collision in Sect. 5. We document our results in Sect. 6. Finally, the conclusion and future work are outlined in Sect. 7.

2 Related works

Our work is closely related to physically correct fracture simulation and meshless methods in computer graphics. To simulate fracture, researchers have proposed varieties of approaches. The early works can date back to Terzopoulos et al. [6], who employed the finite difference method to model and simulate fracture of brittle objects. Furthermore, mass-spring systems [7], the Finite Volume Method (FVM) [8] are presented for deformable solids. Pfaff et al. [9] used dynamically reconstructed mesh to maintain details in cracking simulation of thin sheets. Koschier et al. [10] simulated brittle fracture using tetrahedral mesh with adaptive refinement. The mesh-based methods have proved to be effective for fracture simulation, but the internal force is determined by solving time-consuming linear equations and the high-

quality mesh is hard to deal with. Lots of works concentrate on improving the performance of FEM system [11].

Besides mesh-based methods, many meshless representations are proposed, including level sets [12], particles [13], polyhedra [14] and point clouds [15]. A particle-based numerical solver, namely adaptive SPH, is originated from Bicknell [16]. Shapiro et al. [17] introduced the anisotropic kernel instead of isotropic kernel in traditional SPH. Owen et al. [18] presented an alternative formulation of adaptive SPH algorithm for evolving anisotropic smoothing kernels in cosmological collapse simulation. Recently, Ning et al. [19] used SPH with anisotropic kernel to improve the stability when simulating deformable solids, but their anisotropic axes are fixed. We want to use anisotropic kernel with free axes, and update according to features of solids.

From the geometric viewpoint, fracture means the decomposition of the triangles or tetrahedra. Thanks to the Voronoi diagrams, solutions are proposed to reduce costs based on approximative generation: embedded tetrahedron, virtual node algorithm [20] and fracture mapping [21]. Muller et al. [22] used the Volumetric Approximate Convex Decomposition (VACD) algorithm to deal with visual geometry fracture in real time. Based on implicit function operation, Glondu [23] approximated the deformations of the fragments and dealt with the recursive fracture. Chen et al. [24] utilized a particle representation based on the tetrahedral decomposition. The same goal of the aforementioned work is to find an efficient and stable solution to update the surface when changing topologies. Inspired by [24], our method focuses on a hybrid representation based on both particle and mesh. The mesh is restricted by the physical particles, so that we can preserve the fracture surface quickly.

3 Continuum equation for linear elasticity

Our physical model is mainly motivated by the contact mechanics [25]. When a rigid object deforms, it moves from the original position \mathbf{x}_0 to the current position \mathbf{x}_t , and the displacement field corresponds to the difference between the two positions, described as $\mathbf{u} = \mathbf{x}_t - \mathbf{x}_0$. Thus the Jacobian of the transformation $\mathbf{x}_0 \mapsto \mathbf{x}_t$ is $\mathbf{J} = \mathbf{I} + \nabla \mathbf{u}$ with the identity matrix \mathbf{I} . Using the linear Cauchy-Green strain tensor, the stress ε can be formulated as

$$\varepsilon = \frac{1}{2}(\mathbf{J} + \mathbf{J}^T) - \mathbf{I}. \tag{1}$$

Assuming the rigid body is composed of linear elastic material, the strain σ obeys the Hooke constitutive law $\sigma = \mathbf{C}\varepsilon$. The matrix \mathbf{C} is a 6 by 6 matrix that depends on two constants for isotropic materials: the Young modulus E , and the Poisson ratio ν , while there are twenty-one elastic constants for anisotropic materials.

To deal with the continuous equations, an appropriate numerical method must be employed for model discretization. Due to the efficiency and flexibility of meshless methods, we discretize the object as particles with the SPH model. Ihmsen et al. [26] gave an excellent survey for SPH, where the gradient of displacement field $\nabla \mathbf{u}_i$ of particle p_i is approximated by a smooth function using a finite set of neighboring particles and a kernel function $W(\mathbf{x}_{ij}, h) = W(\mathbf{x}_i - \mathbf{x}_j, h)$:

$$\nabla \mathbf{u}_i = \sum_j V_j \mathbf{u}_{ji} \nabla W(\mathbf{x}_{ij}, h). \tag{2}$$

Here V_i is the volume of p_i , h is the smooth radius, and \mathbf{u}_{ji} denotes the differences between the displacements of neighboring particle p_j and p_i , the corotated approach is used to keep the rotationally invariant, so \mathbf{u}_{ji} is the locally rotated deformation given by

$$\mathbf{u}_{ji} = \mathbf{M}_i^{-1}(\mathbf{x}_j - \mathbf{x}_i) - (\mathbf{x}_{j0} - \mathbf{x}_{i0}), \tag{3}$$

where \mathbf{M}_i is the rotation matrix for p_i . For a detailed introduction, also see [27].

4 Meshless elasticity analysis scheme with anisotropic kernel

As described above, the stress of anisotropic materials is hard to calculate and control owing to complex parameters. Here we introduce the adaptive SPH with anisotropic kernel and the stress analysis scheme based on the elastic model.

4.1 Adaptive SPH with anisotropic kernel

The isotropic kernel $W(\mathbf{x}_{ij}, h)$ can only reflect the inner spacing variations in time and space. It is not suitable for anisotropic deformations, since it does not show any features of the directional effects and will lose neighboring information in some directions. To solve these problems, we employ the adaptive SPH model with an anisotropic kernel, such as ellipsoidal smoothing kernel function. The kernel is characterized by different smoothing length along each axis (see Fig. 1). It can be expressed by a 3 by 3, second order smoothing tensor \mathbf{G} which is real and symmetric. The corresponding anisotropic kernel is

$$W(\mathbf{r}, \mathbf{G}) = \frac{15}{\pi} \|\mathbf{G}\| (1 - \|\mathbf{G}\mathbf{r}\|)^3, \\ \nabla W(\mathbf{r}, \mathbf{G}) = -\frac{45}{\pi} \|\mathbf{G}\| \mathbf{G}^2 \mathbf{r} \frac{(1 - \|\mathbf{G}\mathbf{r}\|)^2}{\|\mathbf{G}\mathbf{r}\|}, \tag{4}$$

where the tensor \mathbf{G} rotates and stretches from the distance vector \mathbf{r} to $\mathbf{G}\mathbf{r}$. Note that, we can treat the isotropic kernel as a special case of the anisotropic kernel when $\mathbf{G} = h^{-1}\mathbf{I}$.

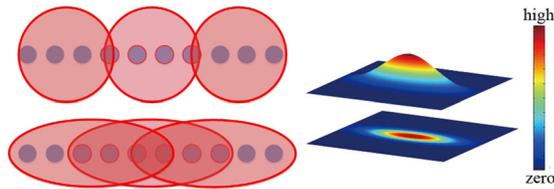


Fig. 1 The kernel pattern of SPH and adaptive SPH in two-dimensional space (*left*) and weight distribution (*right*) of adaptive SPH in three-dimensional space. *Upper left* a metal wire cannot find enough neighbors with isotropic kernel. *Lower left* anisotropic kernel finds more neighbors along the wire

To initialize the anisotropic tensor \mathbf{G} , we decompose \mathbf{G} with the singular value decomposition (SVD):

$$\mathbf{G} = \mathbf{R}\mathbf{G}_k\mathbf{R}^T, \tag{5}$$

where \mathbf{R} is the rotation matrix, whose column vectors correspond to the anisotropic axis directions, and \mathbf{G}_k is a diagonal matrix, the diagonal elements of \mathbf{G}_k denote the inverse smoothing scale along cardinal direction. Given appropriate assignments of \mathbf{R} and \mathbf{G}_k , we can determine the tensor \mathbf{G} directly for elementary geometric objects similar to Fig. 1 according to Eq. (5).

For complex geometric objects, WPCA is suitable to analyze local anisotropy matrix [28], whose corresponding eigenvector of minimal eigenvalue is similar to local normal of surface. The result can be used to enlarge the crack propagation which is perpendicular to the surface. To obtain an invincible surface, just enlarge the minimal eigenvalue and diminish the other two eigenvalues.

To update the anisotropic tensor \mathbf{G} , the simple rotation matrix \mathbf{M} of rigid particles works:

$$\mathbf{G}_t = \mathbf{M}^T\mathbf{G}_0\mathbf{M}, \tag{6}$$

where \mathbf{G}_t denotes the anisotropic tensor at each time step t .

As shown in Fig. 1, assuming that the blue particle is the discrete representation of metal wire which is easy to bend but hard to tear along the wire. When calculating quantities of the blue particles, for isotropic kernel, p_i maybe cannot find enough neighbors, which will lead to simulation breakdown or other unexpected errors. While the anisotropic kernel expands along the wire, we can ensure stability of the simulation and reflect their features in different directions.

Handling the anisotropic features based on adaptive SPH, $\nabla\mathbf{u}_i$ for particle p_i is reformulated as:

$$\nabla\mathbf{u}_i = \sum_j V_j\mathbf{u}_{ji}\nabla W(\mathbf{x}_{ij}, \mathbf{G}). \tag{7}$$

Then, to compute the elastic force of a particle p_i , the strain energy U_i is considered as

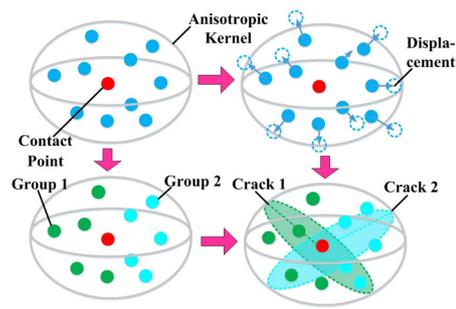


Fig. 2 Procedure of stress analysis scheme. First the analytical point (*red*), and the neighboring fracture particle (*blue*) are calculated. Then the displacement (*virtual particles*) of each neighbor is evaluated by contact mechanics. Finally, we divide the neighbors into two groups, different cracks can be obtained by each group

$$U_i = V_i \frac{1}{2} (\boldsymbol{\varepsilon}_i \cdot \boldsymbol{\sigma}_i), \tag{8}$$

where V_i is the volume of p_i . Assuming that the stress and the strain are constants in the rest volume of each particle, the elastic force \mathbf{f}_{ji} exerted on the particle p_j by p_i can be calculated as:

$$\mathbf{f}_{ji} = -\nabla_{\mathbf{u}_j} U_i = -V_i (\mathbf{I} + \nabla\mathbf{u}_i^T) \boldsymbol{\sigma}_i \mathbf{d}_{ij}, \tag{9}$$

where \mathbf{d}_{ij} is approximated as:

$$\mathbf{d}_{ij} = V_j \nabla W(x_{ij}, \mathbf{G}). \tag{10}$$

According to the above equations, we can determine the elastic behavior and analyze the crack scheme.

4.2 Controllable crack calculation scheme

After the elastic force analysis, a simple scheme is proposed to control the crack generation. We use Rankine hypothesis [29] as fracture criteria. It is stated that if the principal stress of a point exceeds a criterion threshold of the material, a fracture is initiated and the normal of fracture surface equals to the principal stress direction. In this way the scheme can only produce one crack at a point; however, the fracture in the real world produces crack surfaces in different directions, just like cobweb. How to manage the crack propagation and control the number of cracks is still unavailable.

Our method adopts the idea of cluster analysis [30] to tackle the above problem as described in Fig. 2. When collision happens at the red point, we set the anisotropic kernel to equal to the kernel of the nearest particle, then we can obtain a series of fracture points distributed in areas of anisotropic kernel of the contact point. We sort them according to the position and redistribute them into different groups. The number of groups can be aided by interactive parameter setting, for example, in Fig. 2 the number is two. Then, the stress

and strain energy is determined with Eqs. (1) and (8) for each group by averaging its members, producing different cracks. The propagation radii are proportional to the strain energy with the fixed energy release rate G :

$$G = -\frac{U_{a+\Delta a} - U_a}{BS\Delta a}, \tag{11}$$

where U_a means the strain energy when the fracture radius r_{outer} is a , B is the thickness of model, and S is the contact area of collision. Generally speaking, the number of cracks equals to the group number, the procedure is detailed in Algorithm 1. Our stress analysis scheme offers controllability of both crack number and possible extension to special fracture pattern.

The strain energy is supposed to distribute with a radial basis function, when the energy is great enough, the inner radius r_{inner} is introduced. Points in r_{inner} have higher energy than a threshold U_{limit} , we let the particles in the scope of r_{inner} remove their links with neighbors and break into isolated particles to imitate the real scenes. Here, $U_{\text{limit}} = \frac{315}{64\pi h^9}(h^2 - r_{\text{inner}}^2)U$.

5 Simulating fracture with adaptively subdivided particles

After the fracture scheme is established in Sect. 4, we can now produce a high-resolution fracture animation with geometric treatment. Our basic idea is to divide the physical particle adaptively with self-refinement.

Algorithm 1 Crack Generation Scheme

Input: Analyzed Point P , Group Number N
Output: crack normal \mathbf{n} , fracture radius $r_{\text{outer}}, r_{\text{inner}}$
 1: $N_p =$ neighboring particles set of P
 2: $G =$ the nearest neighbor's kernel
 3: **for** $i = 1$ to $\text{card}(N_p)$ **do**
 4: **if** $\text{Gr} > 1$ **then**
 5: Eliminate p_i from N_p
 6: **else**
 7: Calculate displacement \mathbf{u} of p_i
 8: **end if**
 9: **end for**
 10: Sort the elements in N_p by $|\mathbf{r}|$
 11: Divided N_p into N subsets (S_1, S_2, \dots, S_N)
 12: **for all** S_j such that $1 < i < N$ **do**
 13: Calculate $\nabla \mathbf{u}$ with Eq. (7)
 14: Calculate ε, σ, U with Eqs. (1) & (8)
 15: **if** $\lambda_{\text{max}}(\varepsilon) > \text{threshold}$ **then**
 16: $n_i =$ eigenvector for $\lambda_{\text{max}}(\varepsilon)$
 17: Calculate r_{outer} and r_{inner} with Eq.(11)
 18: **end if**
 19: **end for**

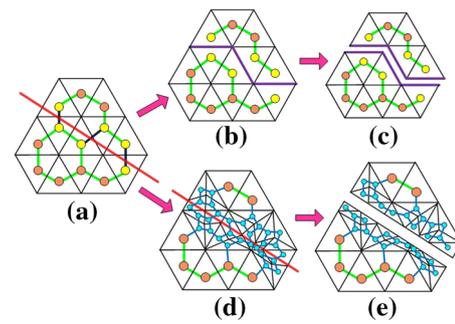


Fig. 3 The procedure of fracture surface generation in two-dimensional space with basic method (b, c) and subdivided particles (d, e). a Links (green) and particles (brown) intersected with the initial crack tip (red) are marked (black line and yellow particles). b Remove the intersected links and generate crack (purple). c Generate new fragments. d Subdivide the corresponding particles (blue) and link them (blue). e Remove the intersected links and generate new fragments

5.1 Physical particle initialization

Inspired by [31], we use the Delaunay-driven tetrahedral generation based on the FEM domain decomposition to approximate model and initialize the physical particles.

First of all, we pre-process the initial object in layered structures due to the formation of anisotropic materials, then use the Delaunay algorithm to generate tetrahedron representation of the model. A particle is allocated to the centroid of each tetrahedron, each particle's mass equals to the mass of tetrahedron and the particle's radius r is measured by its volume $V : r = \sqrt[3]{3V/4\pi}$. Every particle stores two kinds of information: the original tetrahedron vertex positions and the relations between neighboring particles. The original tetrahedral vertices are preserved to maintain the cracking of surface generation and calculate the internal force. The rigid surface can be reconstructed with a constraint between the two neighboring particles to handle the geometric topology modification [31].

5.2 Fracture simulation with adaptively subdivided particles

According to the discrete model described in Sect. 5.1, we can analyze and generate fracture conveniently and quickly. Figure 3b, c illustrates the procedure of basic method for crack generation. First, we find out all the links (the connection between two neighboring physical particles) intersected with the analytical fracture surface. Then we directly remove the intersected links to generate new fragments.

However, the details and the shape of fragments are limited by the initial situation of tetrahedra. In order to preserve the physical and visual realism, the size of tetrahedra should be small enough, but it will have a significant impact on the speed of dynamic computing. So we adopt the idea of

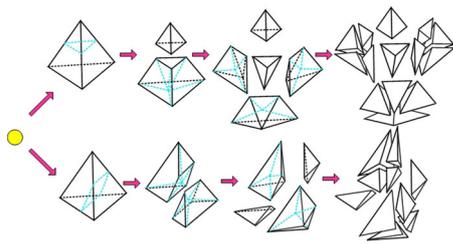


Fig. 4 The particle subdivision scheme for ‘3-1’ (top) and ‘2-2’ (bottom) mode

adaptive subdivision to maintain high resolution in fracture areas and achieve a fast simulation framework.

The new subdivision scheme is shown in Fig. 3d. First, the particles next to the plane are marked, then the tetrahedra belonging to the marked particles are divided into small tetrahedra (particles). In this way we can obtain the analytical result. Then we rebuild relations between the new particles and other particles. In our data structure, one particle has no more than 4 neighbors. A list is maintained to record the extra relations, which will be explained in the subdivided scheme. Finally, we remove the relation in a similar fashion and new fragments with local high resolution are obtained, as shown in Fig. 3e.

By analyzing the situation when a plane intersects with a tetrahedron, tetrahedral vertices have two kinds of relative positions: one vertex on one side of the plane while three on another side; two vertices on each side. We will call them ‘3-1’ and ‘2-2’ mode in the following paragraph.

The algorithm is shown in Fig. 4. Motivated by the idea of Marching Cubes, we mark the tetrahedron vertices with sign ‘+’ and ‘-’, and provide a vertex sign table conserving all 16 possibilities to query. If all vertices are ‘+’ or ‘-’, of course no intersection happens. Realizing the intersection mode, we can subdivide the particle adaptively with the help of vertex list for 14 intersection situations.

For ‘3-1’ mode, we first link the three intersection points with each other and the barycenter of the bottom surface, cutting a top and an inverted tetrahedron off. Then, we link the barycenter with 3 bottom vertices, leaving three rectangular pyramids. The rectangular pyramids can be easily decomposed into 4 tetrahedra by linking the bottom barycenter with other vertices. In this way, a ‘3-1’ particle becomes 14 new particles. Figure 4 provides the detailed process.

An inevitable problem appears that for the original particles the faces in the three vertices side decompose naturally as their neighbors will also intersect with the plane, while the bottom neighbors and the new particles which stem from the bottom will lose neighbors because they do not really share the same surface (see Fig. 5a). So we preserve such relations in a list. When dealing with the relations in this list, we must carefully check all links, but fortunately they are rare cases.

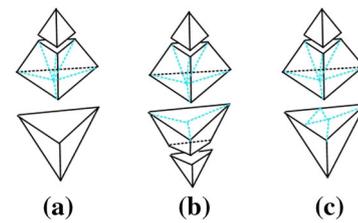


Fig. 5 The situation for extra unbalanced links. **a** The ‘3-1’ subdivision leads the bottom to meet 3 neighbors for one surface. **b** Two neighboring particles are both subdivided in ‘3-1’ mode. If their adjacent surfaces are just not cut by cracks, our method can prove the adjacent surface decompose in the same way (link barycenter with other vertices). So a balanced link is maintained. **c** Unfortunately the neighbor of ‘3-1’ particle is cut at the common surface

For ‘2-2’ mode, the plane cuts a tetrahedron into two triangular prisms, where we select the barycenter of the cutting plane and link it with other vertices. The triangular prism is clearly split into 2 tetrahedra and 2 rectangular pyramids, which can be decomposed as ‘3-1’ mode. So a ‘2-2’ particle evolves into 20 parts. As all neighbors of ‘2-2’ particles will subdivide too, the relevant relations need no more considerations.

If we decompose the particles without any restrictions, the number of particles will explode, because the particles near the analyzed point will be decomposed again and again, so we only carry out the above steps to the particles which have not been subdivided. Another kind of unbalanced links appears. Regardless of ‘3-1’ or ‘2-2’ mode particles, if a cut surface happens to be a previous neighbor of a ‘3-1’ particle’s bottom surface (Fig. 5c), another two kinds (‘3-1’ and ‘2-2’) of new extra relations are stored into the extra relation list.

The volume and mass of new particles can be calculated in line with the original particle’s density, and the velocity is set by the fragment it belongs to.

The last step is necessary to complete the fracture operation. Delete relations containing decomposed original particles, link the new particles correctly based on coplanar situation, so we can detect and generate new surface. The whole process is documented in Algorithm 2. The adaption introduces new degrees of freedom, improving accuracy for fracture tracking and subsequent motion of rigid objects.

5.3 Collision detection with boundary particles

Instead of applying the surface particles as the collision primitive directly, we introduce coarse boundary particles to perform the collision detection. We differentiate the particles into inner particles and surface particles by checking whether it has 4 neighbors or not. Usually the surface particles are employed to detect collision between rigid objects.

For all the surface particles, we use n nearest neighboring particles to generate one boundary particle, where n is the

Algorithm 2 Fracture Surface Generation

```

Input: Surface Point  $P$ , Surface Normal  $n$ , Fracture Radius  $r$ , original
particles set  $P_o$ 
1: for  $i = 1$  to  $card(P_o)$  do
2:   if Distance of particle  $p_i$  to the surface  $d_i < threshold$  then
3:     Find the vertex position in an array  $vtable$ 
4:     if  $vtable_i \in \text{'3-1' or '2-2' mode}$  then
5:       Subdivide  $p_i$  as Fig. 4
6:       if  $p_i$  in  $erlist$  then
7:         Store extra relation in  $erlist$ 
8:       end if
9:       Store extra relation in  $erlist$  if necessary
10:    end if
11:  end if
12: end for
13: Remove the useless old relation in  $rlist$ 
14: Add new relations into in  $rlist$ 
15: Remove the intersected links in  $rlist$ 
    
```



Fig. 6 Boundary particle generation for a pet model. All the surface particles are used to create the boundary particles with a sampling size. *Left* The original surface particles add up to 11228. *Right* 3678 boundary particles for collision detection

user-defined sampling size, it is 4 in our experiments. The mass of a boundary particle equals to the sum of physical particles and the position is the average position. It cannot be ignored that only part of the surface particles will be used for the sampling because the number of boundary particles is not always a multiple of n , the unaided particles located at the sharp corner of the geometry will not have enough neighboring particles. Setting these unaided particles as the collision primitive, the problem can be omitted. An example for boundary particles of a pet model is shown in Fig. 6.

We update the positions of boundary particles that will detect collision at each time step in rigid dynamics. Collision information will be created by two collision boundary particles which belong to different objects. The contact point is chosen to be the analyzed point.

6 Implementation and results

6.1 Implementation

To accomplish our simulation, we use the third-party physical engine “Newton Game Dynamics” (newtondynamics.com) to simulate the movement of rigid bodies and perform collision detection in our scenes. When collision happens, it sends the collision information to the fracture module to analyze

stress. And we generate rendering surfaces from querying the neighboring information of fragments and bodies.

Our C++ code runs on PC with Intel Xeon CPU W3530, 4G RAM memory and NVIDIA GTX760 graphics card. To generate the tetrahedral mesh, we use the NETGEN (sourceforge.net/projects/netgen-mesher). All images are rendered offline by V-Ray (chaosgroup.com).

6.2 Simulation results

Based on the above method, we simulate different fractures on different conditions with different materials. Figure 7 shows a scene of a falling pot. The left one is simulated with the basic fracture surface generation scheme while the right one adopts our method. Obviously, our new adaptive method produces richer details and offers more accuracy for fragments shape and subsequent motion, in contrast, the basic idea will produce artificial polygonal line cracks due to the rough initial mesh.

In Fig. 8, we modify the WPCA kernels in Fig. 7 to reduce/enhance the interaction along the pot surface. It is clear that not only the group number, but also different anisotropic kernels strongly influence the results, it is another effective control facility for user interactivity.

Figure 9 is an example of collapsing glass window. We use a kernel with bigger radius in the direction of window surface to offer enough neighbors. When the collapse cannot break the whole window, our method makes sure the region along the surface stays connected while generating the subtle cracks, the region within the inner radius is destroyed com-



Fig. 7 Falling pot simulation with basic fracture surface generation (*left*) and adaptively subdivided particle method (*right*)



Fig. 8 The same scene of falling pot for anisotropic material with different kernel. The interaction along the surface is adjusted to 1/5 (*left*) and 3 (*right*) times



Fig. 9 A ball collapses through glass window, breaks into a hole and crack propagates without destroying the global structure. Fracture surface normal is the same while different ball sizes lead to different radii. The radii of balls are 3 and 5



Fig. 10 Pots are dropped and fractured into 117 fragments. The *top* two are sideview display and the *bottom* two are zoom-ins from above

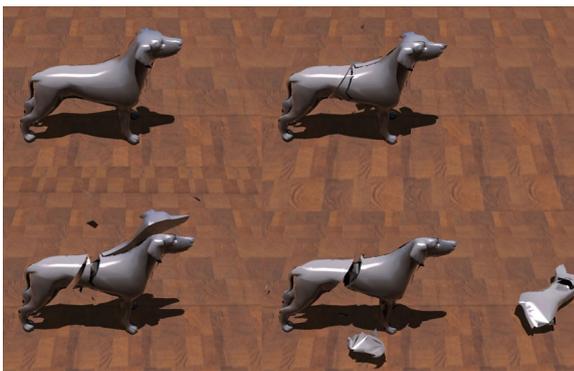


Fig. 11 The pet model impacted by a falling ball. 20k original particles and 30k subdivided particles are created during the simulation

pletely to obtain visual realism. Moreover, the contact area is considered as another factor in this example. When collision happens, we check the contact area of two tetrahedra, which is regarded as a weight in Eq. (11), as shown in Fig. 9, all the balls are initialized in the same way but with different sizes.

To demonstrate the efficiency of handling complex scenarios, Fig. 10 shows the scene of 13 falling pots. Each pot has different neighboring particle groups and anisotropic kernels to produce different cracks. And in Fig. 11, we employ our

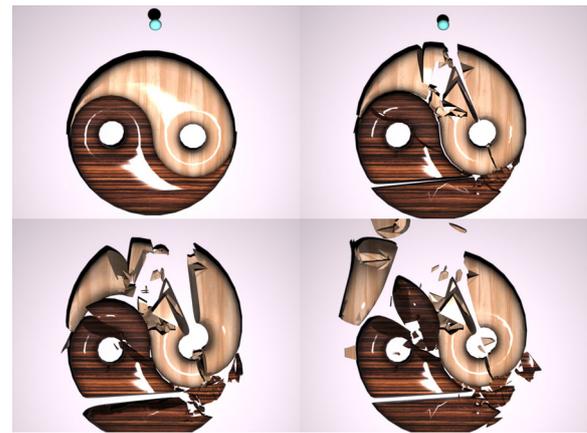


Fig. 12 An ornament of Tai Chi diagram with two kinds of materials, the two parts show different propagation directions during one collision

Table 1 Simulation statistics for the different scenarios

Scenario	Physical particles	Subdivided particles	Boundary particles	Fragments
Pot1	6163	6163	1081	6
Pot2	6163	15717	1081	14
Glass1	1743	9027	314	436
Glass2	1743	9027	314	598
Pots	80,119	144,975	14,053	587
Pet	23,664	57,882	3678	47
Tai Chi	3880	27,278	611	87

Table 2 Computation time statistics for the different scenarios

Scenario	Stress analysis (ms)	Particle subdivision (ms)	Crack generation (ms)	Average frame rate
Pot	86	2869	1505	10.2
Glass	91	1593	329	36.8
Pots	588	12,923	11,037	1.3
Pet	341	86,784	3386	1.2
Tai Chi	649	12,330	4010	7.6

method for the complex modeling of pet. Figure 12 shows the application to rigid objects consisting of hybrid materials. The critical points between different materials are tracked in a list and new fracture are obtained in areas of new materials. Our method can be also extended to multiple materials.

The statistics for quantities in all the examples are shown in Table 1. It can be obviously seen that our physical model has advantages of controlling fracture generation, and the adaptively subdivided particle model offers enhanced details. The number of fragments is greater than what the visual effects could afford as small fragments are ignored in animation. Table 2 documents the performance statistics, the

frame rate is the average number for animation sequences with no less than 200 frames. Compared with other methods, our hybrid method shows efficiency on stress analysis and mesh processing. On the other hand, the subdivision operation is the main bottleneck, but it is amenable for parallel acceleration.

7 Conclusion and future work

This paper has detailed a novel physically based approach to the more accurate simulation of brittle fracture of anisotropic materials. We proposed a new physical model based on elastic continuum mechanics to formulate and deal with the unknown displacement when collision occurs, and employed a new numerical integration based on the anisotropic SPH approximation to solve such continuum equations. Based on the numerical, anisotropic SPH solver, we developed a stress analysis scheme to produce and control cracks in an interactive manner. To improve the visual fidelity, we also presented an adaptive particle model equipped with geometric refinement to ensure both numerical accuracy and visual fidelity of enhanced details. Moreover, extra boundary particles are sampled to ease the burden of computation during collision detection and handling. Our method has exhibited controllability and accuracy in crack simulation and has demonstrated efficiency through our extensive experiments in different scenarios.

There are still some limitations in our novel approach that calls for further improvement in the near future. First, to simulate complex fracture patterns such as ringed and curved cracks, the point of current interest initiating the crack should be determined and managed by a suitable numerical scheme with more precise analysis, this would require us to seek a more adaptable stress analysis scheme. Second, our current method still brings forth some less-desirable artifacts that should be overcome with better and more powerful computational strategies. Third, the generation method for boundary particles in this paper is less ideal, leading to unstable cases during collision handling in some of our experiments due to irregular geometric features of certain models. Finally, our method's performance could be significantly accelerated using the GPU platform in CUDA environment.

Acknowledgments This paper is supported in part by Natural Science Foundation of China (No. 61190120, 61190121, 61190125, and 61272199), National Science Foundation of USA (IIS-0949467, IIS-1047715, and IIS-1049448), National High-tech R&D Program of China (863 Program, No. SS2015AA010504), Innovation Program of Shanghai Municipal Education Commission (No. 12ZZ042), the Specialized Research Fund for Doctoral Program of Higher Education (20130076110008), the open funding project of State Key Laboratory of Virtual Reality Technology and Systems, Beihang University (Grant No. BUAA-VR-15KF-14), and Shanghai Collaborative Center of Trustworthy Software for Internet of Things (No. ZF1213). The authors wish

to thank Dr. Feibin Chen for his technical support on mechanics theory. The authors also wish to thank all the anonymous reviewers for their insightful comments that have helped improve this paper's quality.

References

- O'Brien, J.F., Hodgins, J.K.: Graphical modeling and animation of brittle fracture. In: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, pp. 137–146. ACM Press/Addison-Wesley Publishing Co., New York (1999)
- Zheng, C., James, D.L.: Rigid-body fracture sound with precomputed soundbanks. *ACM Trans. Gr. (TOG)* **29**(4), 69 (2010)
- Pauly, M., Keiser, R., Adams, B., Dutré, P., Gross, M., Guibas, L.J.: Meshless animation of fracturing solids. *ACM Trans. Gr. (TOG)* **24**(3), 957–964 (2005)
- Guo, X., Qin, H.: Meshless methods for physics-based modeling and simulation of deformable models. *Sci. China Ser. F Inf. Sci.* **52**(3), 401–417 (2009)
- Narain, R., Samii, A., O'Brien, J.F.: Adaptive anisotropic remeshing for cloth simulation. *ACM Trans. Gr. (TOG)* **31**(6), 152:1–152:10 (2012)
- Terzopoulos, D., Fleischer, K.: Modeling inelastic deformation: viscoelasticity, plasticity, fracture. In: Proceedings of ACM SIGGRAPH Computer Graphics, vol. 22, pp. 269–278. ACM, New York (1988)
- Levine, J., Bargteil, A., Corsi, C., Tessendorf, J., Geist, R.: A peridynamic perspective on spring-mass fracture. In: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 47–55 (2014)
- Teran, J., Blemker, S., Hing, V., Fedkiw, R.: Finite volume methods for the simulation of skeletal muscle. In: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation, pp. 68–74. Eurographics Association (2003)
- Pfaff, T., Narain, R., de Joya, J.M., O'Brien, J.F.: Adaptive tearing and cracking of thin sheets. *ACM Trans. Gr. (TOG)* **33**(4), 110 (2014)
- Koschier, D., Lipponer, S., Bender, J.: Adaptive tetrahedral meshes for brittle fracture simulation. In: Proceedings of the 2014 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 57–66. Eurographics Association (2014)
- Bao, Z., Hong, J.M., Teran, J., Fedkiw, R.: Fracturing rigid materials. *IEEE Trans. Vis. Comput. Gr.* **13**(2), 370–378 (2007)
- Hegemann, J., Jiang, C., Schroeder, C., Teran, J.M.: A level set method for ductile fracture. In: Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 193–201. ACM, New York (2013)
- Bender, J., Müller, M., Otaduy, M.A., Teschner, M., Macklin, M.: A survey on position-based simulation methods in computer graphics. *Comput. Gr. Forum* **33**(6), 228–251 (2014)
- Martin, S., Kaufmann, P., Botsch, M., Wicke, M., Gross, M.: Polyhedral finite elements using harmonic basis functions. *Comput. Gr. Forum* **27**(5), 1521–1529 (2008)
- Steinemann, D., Otaduy, M.A., Gross, M.: Fast arbitrary splitting of deforming objects. In: Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 63–72. Eurographics Association (2006)
- Bicknell, G., Gingold, R.: On tidal detonation of stars by massive black holes. *Astrophys. J.* **273**, 749–760 (1983)
- Shapiro, P.R., Martel, H., Villumsen, J.V., Owen, J.M.: Adaptive smoothed particle hydrodynamics, with application to cosmology: methodology. *Astrophys. J. Suppl. Ser.* **103**, 269–330 (1996)
- Owen, J.M., Villumsen, J.V., Shapiro, P.R., Martel, H.: Adaptive smoothed particle hydrodynamics: methodology II. *Astrophys. J. Suppl. Ser.* **116**(2), 155–209 (1998)

19. Liu, N., Zhu, F., Li, S., Wang, G.: Anisotropic kernels for meshless elastic solids. In: Proceedings of 12th International Conference on Computer-Aided Design and Computer Graphics (CAD/Graphics), pp. 349–356. IEEE, New York (2011)
20. Molino, N., Bao, Z., Fedkiw, R.: A virtual node algorithm for changing mesh topology during simulation. In: Proceedings of ACM SIGGRAPH 2005 Courses, pp. 4–11. ACM, New York (2005)
21. Desbenoit, B., Galin, E., Akkouche, S.: Modeling cracks and fractures. *Vis. Comput.* **21**(8–10), 717–726 (2005)
22. Müller, M., Chentanez, N., Kim, T.Y.: Real time dynamic fracture with volumetric approximate convex decompositions. *ACM Trans. Gr. (TOG)* **32**(4), 115–124 (2013)
23. Glondu, L., Marchal, M., Dumont, G.: Real-time simulation of brittle fracture using modal analysis. *IEEE Trans. Vis. Comput. Gr.* **19**(2), 201–209 (2013)
24. Chen, F., Wang, C., Xie, B., Qin, H.: Flexible and rapid animation of brittle fracture using the smoothed particle hydrodynamics formulation. *Comput. Anim. Virtual Worlds* **24**(3–4), 215–224 (2013)
25. Johnson, K.L., Johnson, K.L.: *Contact Mechanics*. Cambridge University Press, Cambridge (1987)
26. Ihmsen, M., Orthmann, J., Solenthaler, B., Kolb, A., Teschner, M.: Sph fluids in computer graphics. In: Proceedings of Eurographics (State-of-the-Art Report), pp. 21–42. The Eurographics Association (2014)
27. Becker, M., Ihmsen, M., Teschner, M.: Corotated sph for deformable solids. In: Proceedings of NPH, pp. 27–34. Citeseer, Munich (2009)
28. Yu, J., Turk, G.: Reconstructing surfaces of particle-based fluids using anisotropic kernels. *ACM Trans. Gr. (TOG)* **32**(1), 5:1–5:12 (2013)
29. Gross, D., Seelig, T.: *Fracture Mechanics: with an Introduction to Micromechanics*. Springer, Berlin (2011)
30. Estivill-Castro, V.: Why so many clustering algorithms: a position paper. *ACM SIGKDD Explor. Newsl.* **4**(1), 65–75 (2002)
31. Smith, J., Witkin, A., Baraff, D.: Fast and controllable simulation of the shattering of brittle objects. *Comput. Gr. Forum* **20**(2), 81–91 (2001)



Chen Li is currently a Ph.D. student at Software Engineering Institute, East China Normal University, China. He received his B.E. degree in computer science from Tianjin University in 2013. His research interests include physically based animation and virtual reality.



information visualization, et al.

ChangBo Wang is a professor of Software Engineering Institute, East China Normal University, P.R. China. He received his Ph.D. degree at the State Key Lab. of CAD & CG, Zhejiang University in 2006, and received B.E. degree in 1998 and ME degree in civil engineering in 2002, respectively, both from Wuhan University of Technology. His research interests include physically based modeling and rendering, computer animation and realistic image synthesis, information



Modeling, Computer Aided Design, Computer Aided Geometric Design, Computer Animation and Simulation, Virtual Environments and Virtual Engineering, etc.

Hong Qin is a Full Professor of Computer Science in Department of Computer Science at State University of New York at Stony Brook (Stony Brook University). He received his B.S. (1986) degree and his M.S. degree (1989) in Computer Science from Peking University in Beijing, China. He received his Ph.D. (1995) degree in Computer Science from the University of Toronto. His research interests include Computer Graphics, Geometric and Physics-based