COMPUTER ANIMATION AND VIRTUAL WORLDS
*Comp. Anim. Virtual Worlds* 2013; **24**:215–224
Published online 3 May 2013 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/cav.1514

SPECIAL ISSUE PAPER

# Flexible and rapid animation of brittle fracture using the smoothed particle hydrodynamics formulation

Feibin Chen[1], Changbo Wang[2]*, Buying Xie[1] and Hong Qin[3]

[1] College of Civil Engineering, Tongji University, Shanghai, China
[2] Software Engineering Institute, East China Normal University, Shanghai, China
[3] Department of Computer Science, Stony Brook University, Stony Brook, USA

## ABSTRACT

This paper presents a hybrid animation approach to the flexible and rapid crack simulation of brittle material. At the physical level, the local stress tensors induced by collision are analyzed by using the smoothed particle hydrodynamics (SPH) formulation. Specifically, in order to determine the internal stress when rigid bodies collide with each other or neighboring environments, we treat all of them as completely rigid body that has infinite stiffness and then evaluate virtual displacement for colliding particles. At the geometric level, in order to faithfully maintain the fracture interface during the crack simulation, we utilize an efficient shape representation of solid based on the tetrahedral decomposition of the original solid geometry. This novel hybrid approach resorts to local particle models, whose goal is to avoid heavy computational burden during crack interface updating and topological changing, and meanwhile, it facilitates the user-initiated interactive control during the crack generation and propagation. Our animation experiments demonstrate the effectiveness of our novel particle-based method to simulate the crack of brittle material. Copyright © 2013 John Wiley & Sons, Ltd.

**KEYWORDS**

animation; smoothed particles hydrodynamics (SPH); brittle fracture

***Correspondence**

Changbo Wang, Software Engineering Institute, East China Normal University, Shanghai, China.
E-mail: cbwang@sei.ecnu.edu.cn

## 1. INTRODUCTION

In our everyday life, brittle fracture such as shattering glass and collapsed brick wall as natural phenomena occurs frequently. In the areas of civil engineering and material sciences, the research on studying fracture has a long history, and different kinds of technical approaches have been proposed. In contrast, the effective simulation of cracking material in computer animation was initiated just over one decade ago, with limited successes. It may be noted that computer animation emphasizes visual effects and plausible behaviors much more than accurate numerical results that are universally accepted in civil engineering and material sciences. Therefore, model simplifications and numerical optimization/acceleration are far more favorably accepted in computer animation, as long as visual results are not sacrificed.

This paper proposes a hybrid animation approach to the flexible and rapid crack simulations of brittle material. In principle, there are two technical challenges when simulating the brittle fracture. First, identifying the exact location and orientation of the fracture and managing the propagation of crack is a difficult task, which desperately calls for better computational models. Second, although simulating brittle fracture, always maintaining a legitimate mesh that is conforming to fracture initialization and propagation is a notoriously difficult task that unavoidably evolves frequent topology changes and consumes much more computational time.

Recent years have witnessed the widespread use of physically based numerical simulation to handle fracture. Such physics-based approaches have given rise to a great achievement so far, as it is mainly based on computational fracture mechanics and material mechanics. Using different numerical discretized approaches, the stress caused by the impact of two bodies can be determined precisely. Two types of discretizations exist in the simulation of fracture material: the mesh-based method and meshless method. The finite element method (FEM) is frequently used as a representative of mesh-based approaches. The FEM typically decomposes the original model into a mesh and computes the node property in the mesh using the neighboring mesh element. The predominate difficulty associated with the FEM is that it required a high-quality mesh, which is hard to obtain and it is even more costly to update wherever fracture occurs. In contrast, meshless methods have

been developed in recent years; and they are essentially particle-based or point-based approaches to simulate the behavior of material while eliminating the limitations in the mesh-based methods.

In this paper, we develop a novel hybrid simulation technique for animating fracture of brittle material, such as glass and porcelain. In a nutshell, we propose a smoothed particle hydrodynamics (SPH) formulation to accurately evaluate material fracture and determine crack propagation. As a result, our SPH method is employed to tightly couple with the original solid geometry, determine the initial fracture caused by rigid body collision, and accurately track the crack interface, this way, we expect to significantly reduce the computational cost during the topology updating for the underlying mesh. This paper's main contributions are as follows.

- We adapt the SPH formulation into linear elastic mechanical model and synchronize the SPH formulation with mesh-based solid geometry. To analyze the strain and stress caused by collision, we tackle a virtual displacement problem for impacted particles, and our hybrid method is more convenient than both the FEM and other meshless methods.
- We propose a simple yet effective shape representation scheme for crack interface updating and stress computation. Our method tracks the fracture surface by updating the particle state, which significantly reduces the computational cost in mesh construction and topology changing when comparing with the previous meshless methods.
- We introduce a novel adaptive sampling method based on dynamic position updating to improve the accuracy for processing cracks and managing the number of fragments via user control. Our framework can handle brittle fracture with improved physical accuracy, flexible user control, and reduced computational cost.

## 2. RELATED WORK

From a physical point of view, earlier works to simulate deformation and fracture can date back to [1,2], where the authors pioneered the use of physically based method for animating elastic deformation and plastic fracture in graphics. They made use of finite difference schemes to solve partial differential equations. Mass-spring [3] or mass-constraint [4] models are also popularly used to simulate fracture because of their convenience and efficiency. The limitation of the aforementioned methods is that they cannot evaluate the location and orientation of fracture exactly. The more accurate models for a fracture system are based on the continuum mechanics [5]. As a type of mesh-based discretization methods, FEM is first introduced to solve the continuum mechanics in simulation of brittle and ductile fracture by O'Brien *et al.* [6,7], where only the linear elastic fracture mechanics is utilized. Other works that employ the FEM on fracture are focusing on reducing computational inaccuracy caused by the large time step [8], and improving the performance of the simulation system [9,10]. The deficiency of mesh-based methods is the requirement of large-scale meshes when precise results are needed. Meshless methods, on the other hand, do not have such problem. Pauly *et al.* [11] introduced the moving least squares (MLS) to calculate the strain tensor in their animation system of elastic and plastic fractures. One shortcoming of MLS is that sufficiently large number of neighboring particles is necessary to guarantee that the motion matrix is invertible. Liu *et al.* [12] adapted the meshless local Petrov–Galerkin (MLPG) to solve this problem. They analyzed the stress by solving a quasistatic problem. However, the computational time of solving a quasistatic MLPG formulation is rather high. More convenient and flexible meshless numerical methods are still needed to simulate fracture with high efficiency.

Meanwhile, various methods from the geometric viewpoint have been proposed to track/manage fracture interface while obtaining visually plausible results. There mainly exist two types of surface models: the explicit and implicit models. Explicit model has low computational cost but lacks variability in general, such as embedded tetrahedral meshes [9] and regularly shaped cube meshes [13], which avoid frequent updating in the original mesh. The virtual node algorithm [14] proposed by Molino *et al.* allows element duplication in the crack process to supply abundant meshes. Later on, a novel explicit surface tracking method [15] is presented to split meshless deforming solids fast and efficiently, which combines the advantages of explicit surface with the flexibility of a meshless discretization of the underlying deformation field. In contrast, implicit model can handle complicated situation but has low computing performance. A point-based implicit tracking surface model is presented in [10], where the crack surface is represented implicitly using constructive solid geometry operations. The trend of the aforementioned work is to find an efficient and stable solution to update the surface when changing topology.

## 3. PHYSICAL SOLID MODEL WITH PARTICLES

This section first introduces the linear elastic mechanical model in continuum mechanics that enables deformation and fracture, and then the continuum model is discretized by an SPH formulation for the convenience of computer simulation.

### 3.1. Linearly Elastic Solid Modeling

The physical model of elastic solid in our framework is based on continuum mechanics. A detailed introduction to this field can be found in [5]. Let $\mathbf{x_0} = [x_0,\ y_0,\ z_0]^T$ and $\mathbf{x}_t = [x_t,\ y_t,\ z_t]^T$ define the original and current positions in rigid body. $\mathbf{u} = [u,\ v,\ w]^T$, which denotes the displacement vector field in a deformed state, $\mathbf{x}_t = \mathrm{x}_0 + \mathrm{u}$. The deformation can be measured by the


DOI: 10.1002/cav

gradient $\nabla\mathbf{u}$. Let $\mathbf{J}$ denote the Jacobian of the mapping $\mathbf{x}_0 \rightarrow \mathbf{x}_t : \mathbf{J} = \nabla\mathbf{x}_0 + \nabla\mathbf{u}^T$. Using the linear Cauchy–Green strain tensor, we can define the strain $\boldsymbol{\varepsilon}$ as

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

Assume that the rigid body is approximated by the linear elastic material, the stress $\boldsymbol{\sigma}$ is linearly calculated by $\boldsymbol{\sigma} = \mathbf{C}\boldsymbol{\varepsilon}$, where

$$\mathbf{C} = D\begin{pmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{(1-2\nu)}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{(1-2\nu)}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{(1-2\nu)}{2} \end{pmatrix} \tag{2}$$

where $D = E(1+\nu)(1-2\nu)$, and $E$ and $\nu$ are the Young's modulus and Poissons' ratio of the material, respectively.

### 3.2. Smoothed Particle Hydrodynamics Discretization

By using the continuous model introduced in Section 3.1 earlier, we can simulate the material behavior with an appropriate numerical method through discretization. SPH is employed in our system because of its efficiency and flexibility in numerical implementation [16]. It may be noted that earlier work has already shown the SPH potential for modeling of solid deformation and fracture [17].

Smoothed particle hydrodynamics was developed in the late 1970s [18] and started to gain popularity in the early 1990s. In SPH, the value of a continuous function $f(x_i)$ at a position $x_i$ is approximated by a smooth function $\langle f(x_i)\rangle$ using a finite set of particles and a kernel function $W(x_i - x_j, h)$ with varying radius $h$. According to Gingold and Monaghan [19], the original formulation of the SPH is

$$\langle f(x_i)\rangle = \sum_j \frac{m_j}{\rho_j} f(x_j) \cdot W(x_i - x_j, h) \tag{3}$$

where $m_j$, $\rho_j$ are the mass and density of the particle, respectively.

### 3.3. Displacement Field

According to the aforementioned SPH formation, the elastic model can be discretized. For each particle $\mathbf{x}_i$ in the rigid body, the gradient of the displacement field $\nabla\mathbf{u}_i$ described in Section 3.1 can be evaluated by

$$\nabla\mathbf{u}_i = \sum_j \frac{m_j}{\rho_j}\mathbf{u}_{ji} \cdot \nabla W(\mathbf{x}_i - \mathbf{x}_j, h) \tag{4}$$

where the vector $\mathbf{u}_{ji}$ denotes the differences between the displacement vector $\mathbf{u}_j$ of the neighboring particle $j$ and $\mathbf{u}_i$ of the current particle $i$. It can be computed by

$$\mathbf{u}_{ji} = \mathbf{u}_j - \mathbf{u}_i = (\mathbf{x}_{jt} - \mathbf{x}_{it}) - (\mathbf{x}_{j0} - \mathbf{x}_{i0}) \tag{5}$$

In [20–22], the corotated approach is used to model the material deformation. Its aim is to keep the rotationally invariant when using the linear Cauchy–Green strain tensor to analyze the deformation. Unlike the deformation modeling, the solid is treated as infinitely rigid in our system. The location of a rigid body is described by two parameters in rigid body dynamics: translation and rotation. The rotation matrix of the whole rigid body can be used to perform the rotation modification. Taking the mass center of the rigid body $\mathbf{x}_{ct}$ at time $t$ as a reference point, we can obtain $\mathbf{x}_{jct} = \mathbf{x}_{jt} - \mathbf{x}_{ct}$, $\mathbf{x}_{ict} = \mathbf{x}_{it} - \mathbf{x}_{ct}$, and $\mathbf{x}_{jt} - \mathbf{x}_{it} = \mathbf{x}_{jct} - \mathbf{x}_{ict}$. Using the inverse of rotation matrix $\mathbf{R}^{-1}$ in rigid body, we can replace Equation (5) by using

$$\mathbf{u}_{ji} = \mathbf{u}_j - \mathbf{u}_i = \mathbf{R}_t^{-1}(\mathbf{x}_{jct} - \mathbf{x}_{ict}) - (\mathbf{x}_{jc0} - \mathbf{x}_{ic0}) \tag{6}$$

### 3.4. Virtual Displacement Evaluation

After calculating the internal stress of the material induced by the external impact by the aforementioned formulated displacement vector field, a set of particles are used to represent the solid, the collision that happened in particles naturally gives rise to the stress, because our solid is treated as a completely rigid body so that a virtual displacement of collided particles must be determined before calculating the internal stress.

When a collision occurs, the collision information of a particle is used to compute a deformed state for stress computation. As shown in Figure 1, a virtual displacement state for the colliding particle can be evaluated according to the penetrate depth. After the virtual displacement is resolved, the gradient and stress are calculated by the SPH formulation detailed in Section 3.3.

## 4. SHAPE GEOMETRY AND SURFACE MODEL

The model described in Section 3 can solve the brittle fracture at the physical level. Another important task when

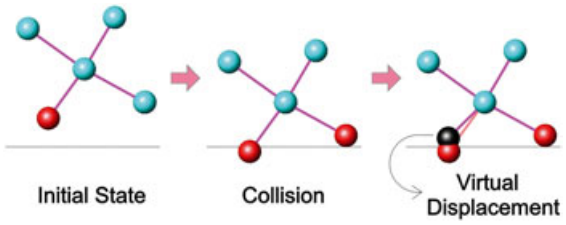


**Figure 1.** Compute the virtual displacement when collision occurs. The position of virtual particle (colored in black) depends on the penetrating depth of the collision particle.

simulating cracking is the shape geometry. Suitable surface model will reduce the burden in crack process and is expected to produce high-quality visual effect. Compared with the high accuracy of implicit surface model [11], an explicit surface model with particles is developed in this paper, and our goal is to obtain a faster computational speed.

In [23], rigid body is discretized by a series of 3D grid, and each voxel is assigned one identical particle. We adapt the similar idea of solid model [4] in the surface representation and append a tetrahedron-based method to distribute these particles.

Figure 2 shows the procedure for the solid discretization. First, the rigid body is decomposed into tetrahedra. Then, a particle is assigned to the barycenter of each tetrahedron. The radius $r$ of a particle is calculated by the volume $V$ of tetrahedron $r = \sqrt[3]{3V/4\pi}$.

For each particle, two types of information must be stored: point geometry and four pointers to its immediate neighboring particles. The geometric points (blue points in Figure 3(a), which inherit the four vertices of original tetrahedron, have two functions as follows: maintaining the crack surface and computing the internal force. The pointer to neighboring particles (The 'link' in Figure 3(a)) indicates the link of a neighboring tetrahedron in the original mesh generation. Two types of particles are defined by its number of neighboring particles: inner and surface particles (Figure 3(a)). The surface particle is used for detecting collision in rigid dynamics, and the triangular face not penetrated by the 'link' will be added to a mesh for rendering. These two types of information are necessary to model the propagation of fracture and generate triangular surface for fragment generation. More detailed discussion is in Section 5.

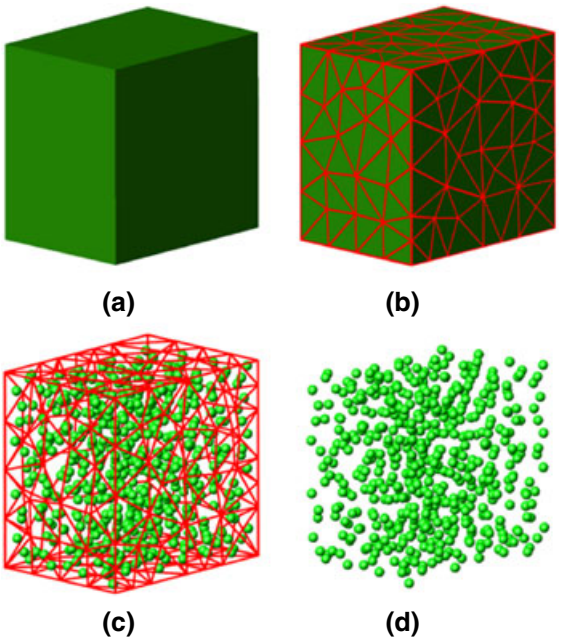


**Figure 2.** Procedure to bridge shape geometry and smoothed particle hydrodynamics particles. (a) Original solid model. (b) Using tetrahedra to decompose the solid model. (c) Assign particles to the barycenters of tetrahedra. (d) The solid is converted to particles.

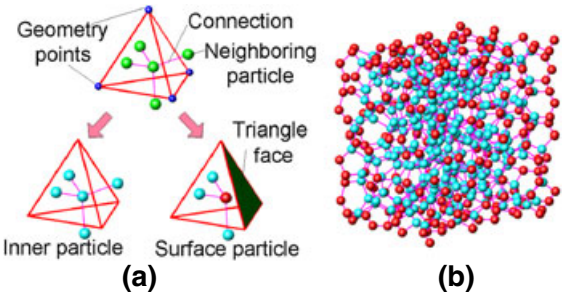


**Figure 3.** Using mesh information to classify particles into two types. (a) An original particle (green) is defined as surface particle (red, has less than four neighbors) or inner particle (cyan, has four neighbors). Blue points are the geometry points and carmine lines denote the links, the triangular face belonging to the surface particle is used for rendering. (b) The whole solid is abstracted by particles and 'links'.

After the aforementioned definition, the whole solid is represented by a cluster of particles and a number of 'links' (Figure 3(b)). And they can be dynamically updated when crack occurs and starts to propagate. We shall also emphasize that this is not the only strategy to divide the solid into tetrahedra; other decomposition schemes such as hexahedra or cubes are also possible as long as the particle can be designed to inherit all the required information that can better manage crack and its propagation.

## 5. CRACK IDENTIFICATION AND PROPAGATION

When the material deformation is exceeding certain threshold, crack will inevitably happen. How to find the initial position where crack starts to occur and how to track/simulate the crack propagation shall be handled by the following computational steps.

To identify fracture initialization, a fracture criteria must be chosen first. The Rankine criteria [24], which is satisfactory for brittle materials is adapted. Fracture occurs when the principal component $p_{max}$ of the stress $\boldsymbol{\sigma}$ is above a certain material threshold $p_m : p_{max} > p_m$, and the normal of fracture surface is the corresponding vector's direction. As we mentioned in Section 4, the geometry points of each particle are our initial sampling positions to compute the stress by using the SPH formation described in Section 3.2. However, employing these initial sampling points to analyze the internal stress directly will not guarantee the accuracy of computation in high stress area. And how to manage the crack propagation and the number of fragments is still a technical challenge in computer graphics nowadays. We now propose an adaptive


DOI: 10.1002/cav

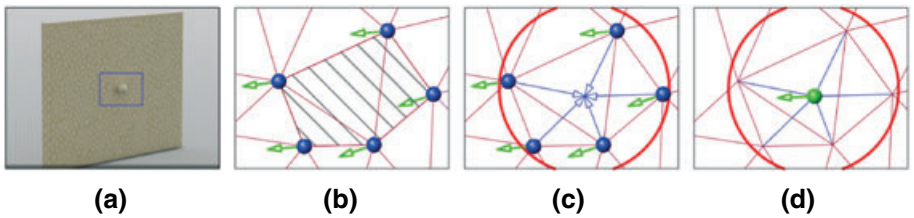


**Figure 4.** Local stress analysis based on the position average scheme. (a) The collision occurs in the identified area. (b) Original fracture points (colored in blue) in the high stress area (shaded by line segments) calculated by fracture criteria. (c) Use the average position and normal (green arrow) to resample the particle, all fracture points have similar normal within an area (red circle) are gathered to calculate a new fracture point (colored in green in (d)).

sampling method based on the dynamic position average scheme. This scheme is inspired by the cluster analysis [25] and aims to tackle the problem of crack identification and propagation.

### 5.1. Stress Analysis Based on Position Average Scheme

Unlike the remeshing method adopted by FEM [6] or iterative resampling used by other meshless methods [15], our method makes use of the idea of cluster analysis [25]. Upon the initial computation of all the geometry points, a series of fracture points that satisfy the fracture criteria have been obtained. They are distributed in areas after the initial evaluation, as shown in Figure 4(b). Some of the initial fracture points are close by according to their fracture position and normal. First, the fracture points are sorted according to its position and the stress normal, and then a user-defined sample radius is introduced to redistribute the fracture points into different groups, points having similar stress normal nearby the sample radius are gathered into one group. Then, a sample point of each group is calculated. The position and the stress normal of the sample point is the average value of all points in the same group, as shown in Figure 4(c–d).

The dynamically sampling points are the final locations used to create solid fracture. This adaptive sampling method can be aided by interactive parameter setting. Moreover, using the sample radius to recalculate the sample fracture point has another utility, as it helps the user to control the number of fragments. The smaller value of the sample radius means that the more sample fracture points would be produced, and a larger sample radius results in a smaller number of fragments.

### 5.2. Fracture Handling Through Link Decomposition

Using the surface model presented in Section 4, we store at most four links in one particle. The fracture process is mainly focused on handling the links between SPH particles. To simulate and manage crack propagation, we introduce a fracture radius to control the influence of the fracture point. When a fracture point is determined, we first query all the links located in the spherical region enforced by the radius and then handle these links related to the fracture plane. Any link that intersects with the fracture plane will be removed and two relevant particles are marked as surface particles, and two triangles representing a part of fracture surface is added to the mesh for rendering. Note that the fracture radius has the same function of sample radius, which is another parameter aiding users to control the fracture. Figure 5 illustrates the overall fracture process, and Figure 6 demonstrates the detailed steps of handling one link through decomposition.

To summarize, the whole fracture procedure can be achieved by the following four steps:

(1) Compute the maximal eigenvalue of stress tensor at all geometry points. If the largest eigenvalue of stress tensile exceeds a material threshold; we use the stress vector as the normal of fracture plane. All the geometry points satisfying the fracture criteria will be collected as a set of all potential fracture points.
(2) Use the position average scheme (detailed earlier) to dynamically compute the new sample points from all potential fracture points.

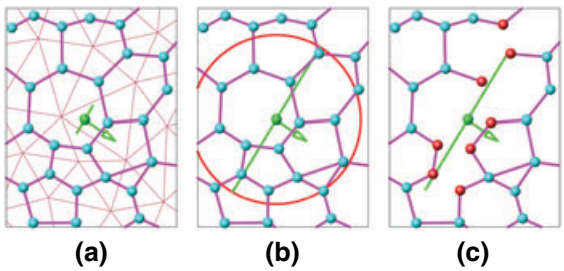


**Figure 5.** Fracture handling. (a) An initial fracture point (colored in green) and its normal (green arrow) in the solid. (b) Using a fracture radius (red circle) to gather all the 'links' within the area. (c) Tracking crack interface, any 'link' intersects with the fracture plane (green line) is removed, and relevant particles are set to surface particles (colored in red).

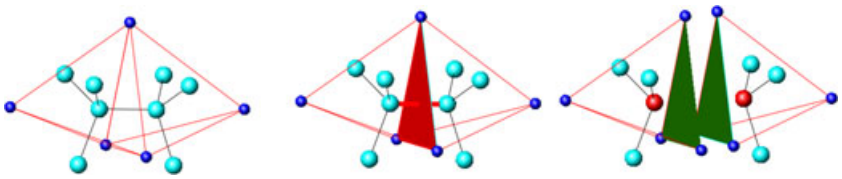

**Figure 6.** Remove one link when fracture occurs. Left: initial state. Middle: a fracture plane intersects with the link (red line). Right: mark the two particles belonging to the link as surface particles and create new triangles to track crack interface.

(3) Remove the links using the sample points as the initial fracture point and a defined fracture radius to manage the crack propagation. For each removed link, modify the relevant particles accordingly. Add new triangles to the rendering mesh.
(4) Perform a group search to cluster the particles in different fragments by a Depth-First Search or Breadth-First Search algorithm, then re-initialize each fragment as a new rigid body.

## 6. IMPLEMENTATION DETAILS AND EXPERIMENTS

### 6.1. Implementation

Figure 7 illustrates the functional pipeline of our simulation system. At each time step, a solid model is governed by rigid body dynamics. The collision detection is performed by surface particles, and a k-dimensional tree, which is rebuilt once at each time step and is used to manage the entire surface particles, only these surface particles belong to different rigid bodies will be used to maintain contact information. The impulse-based method presented by Bender [26] is used here to handle the collision response. An impulse is calculated for each contact point in an iterative loop and apply to the two colliding rigid bodies. In the analysis of strain and stress, particles in rigid bodies are managed by another KD-tree to accelerate neighbor searching for SPH computation. Spiky smoothing kernel function [27] is employed here. All the points satisfying the fracture criteria are used to compute the sample points using the adaptive position average scheme. After that, the sample fracture points are used to update the mesh and generate fragments.

Our C++ code runs on PC with 3.09 GHz CPU, NVIDIA GT460 graphics card and 4 G RAM memory. To generate the tetrahedral mesh and create the data structure for surface model, we use the TetGen (tetgen.berlios.de). All images in our experiments are rendered offline by V-Ray (www.chaosgroup.com).

### 6.2. Animation Results

Figure 8 shows a falling cylinder-shaped pot with different sample radii. In these examples, the sample radii of Figure 8(b–d) are 0, 50, and 150 mm, respectively. It can be obviously observed that by using different sample radii, the number of fragments is convenient to control.

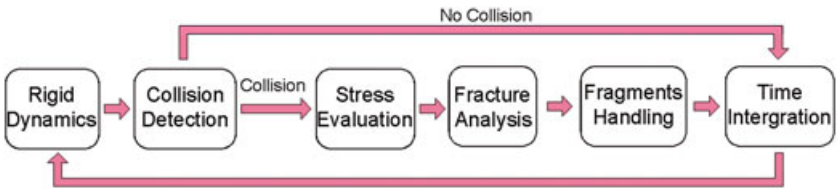


**Figure 7.** The pipeline of our simulation process with different functionalities.


**Figure 8.** A cylinder pot falls onto the ground. Different sample radii are used to control the number of fragments. (a) Initial state. (b) No sample radius control. (c) Result from sample radius 50 mm. (d) Result from sample radius 150 mm.


DOI: 10.1002/cav

Figures 9 and 10 show an example of stone ball hitting a wall. The fracture radius is another parameter to help users to manage and control the fracture simulation. In Figure 9(b), the fracture radius is 300 mm, the colliding ball only generates a hole in the wall. As the fracture radius increases to 600 mm in Figure 9(c), the damage is getting more extensive. In Figure 9(d), the entire wall is almost collapsed by using the fracture radius 1200 mm.

To illustrate the proper handling of complex scenarios, Figure 11 shows that the 'CASA' Logo collides with another Logo. Here, each letter has different control fracture radii and sample radii to produce different results.

The performance statistics for all the examples are shown in Table I. The comparison between our work and related work are documented in Table II. Compared with other methods such as FEM, MLS, or MLPG, our hybrid method has clear advantages on crack propagation, mesh processing, and stress analysis. It only needs an extra preprocess of the original model and the computation overhead in dynamic update (including crack propagation and management) is very low.


**Figure 9.** A ball collides with a clay wall. To control the fracture propagation, different fracture radii are employed. (a) Initial state. (b) Result from fracture radius 300 mm, where only a hole on the wall is generated. (c) Result from fracture radius 600 mm, where the damage is more extensive. (d) Result from fracture radius 1200 mm, where the entire wall is collapsing.

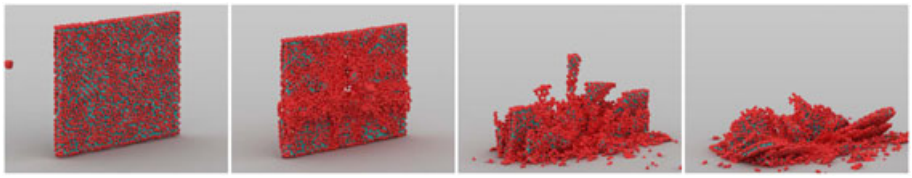

**Figure 10.** Collapsed wall rendered by particles, where red particles are the surface particles and inner particles are colored in cyan.


**Figure 11.** The CASA Logo is colliding with each other, where each character has different sample radii (210 mm on average) and fracture radii (1400 mm on average) to control the simulation results.

**Table I.** The performance statistics for our simulation examples.

| Scene | Rigid | Total particles | Fragments | Fracture radius (mm) | Sample radius (mm) | Stress analysis (ms) | Crack production (ms) |
|---|---|---|---|---|---|---|---|
| Falling pot1 | 1 | 5393 | 51 | 150 | 0 | 104 | 41 |
| Falling pot2 | 1 | 5393 | 18 | 150 | 50 | 112 | 22 |
| Falling pot3 | 1 | 5393 | 12 | 150 | 150 | 77 | 20 |
| Wall 1 | 1 | 7904 | 160 | 300 | 750 | 141 | 44 |
| Wall 2 | 1 | 7904 | 408 | 600 | 750 | 138 | 79 |
| Wall 3 | 1 | 7904 | 719 | 1200 | 750 | 140 | 126 |
| CASA logo | 8 | 28554 | 528 | 200 (average) | 1400 (average) | 423 (total) | 210 (total) |

**Table II.** The comparison between pervious work and our method.

| Related work | Rigid model | Fracture pattern | Numerical model | Analytical efficiency | Surface update | Fragment complexity |
|---|---|---|---|---|---|---|
| Bao[8] | Mesh | Brittle | FEM | Slow | Slow | Medium |
| Pauly[11] | Particle | Ductile & Brittle | MLS | Medium | Medium | Medium |
| Liu[12] | Particle | Brittle | MLPG | Medium | Medium | Simple |
| Ours | Particle | Brittle | SPH | Medium | Fast | Abundant |

FEM, finite element method; MLS, moving least squares; meshless Local Petrov–Galerkin; SPH, smoothed particle hydrodynamics.

## 7. CONCLUSION AND FUTURE WORK

We have developed a hybrid animation approach to the flexible and rapid crack simulation of brittle material. An SPH formulation is adapted to solve the linear elastic mechanics for fracture animation at the physical level. Virtual displacement of colliding particles is calculated to determine the gradient field of displacement when rigid bodies start to collide. From the geometric standpoint, an improved tetrahedron-based surface model is introduced to reduce the computation cost while maintaining the crack surface. The fracture and sample radii are the only two parameters for users to fine-tune their animation results. Our method exhibits low computation burden and has advantage in mesh processing and crack propagation.

There are some limitations in our framework that call for immediate improvement in the near future. First, to create crack surface with more details, the original mesh size must be small, and it will consume more system memory and processing time during collision detection. Second, the fragment evolution needs to handle a large number of rigid body movement, and it is a bottleneck for any simulation system when a large number of objects interact with each other. Another possible improvement is to use the adaptive-sampling scheme to generate small particles in the fracture process dynamically in the interest of more realistic results. Our particle-based method could also be migrated onto the GPU platform.

## ACKNOWLEDGEMENTS

This paper was partially supported by the Natural Science Foundation of China (Grant No. 61070128, 61272199), the National Natural Science Foundation of China (Grants No. 61190120, 61190121, and 61190125), the National Science Foundation of USA (Grants No. IIS0949467, IIS1047715, and IIS1049448), the Innovation Program of the Shanghai Municipal Education Commission (Grant No. 12ZZ042), the Fundamental Research Funds for the Central Universities, and Shanghai Knowledge Service Platform for Trustworthy Internet of Things (Grant No. ZF1213).

## REFERENCES

1. Terzopoulos D, Platt J, Barr A, Fleischer K. Elastically deformable models, In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, Anaheim, California, USA, 1987; 205–214.
2. Terzopoulos D, Fleischer K. Modeling inelastic deformation: viscolelasticity, plasticity, fracture, In *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques*, Atlanta, Georgia, USA, 1988; 269–278.
3. Norton A, Turk G, Bacon B, Gerth J, Sweeney P. Animation of fracture by physical modeling. *The Visual Computer* 1991; **7**(4): 210–219.
4. Smith J, Witkin A, Baraff D. Fast and controllable simulation of the shattering of brittle objects. *Computer Graphics Forum* 2001; **20**(2): 81–90.
5. Fung YC. *A First Course in Continuum Mechanics*. Prentice-Hall, Englewood Cliffs, N.J, 1969.
6. O'Brien JF, Hodgins JK. Graphical modeling and animation of brittle fracture, In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, Los Angeles, California, USA, 1999; 137–146.
7. O'Brien JF, Bargteil AW, Hodgins JK. Graphical modeling and animation of ductile fracture. *ACM Transactions on Graphics* 2002; **21**(3): 291–294.
8. Bao Z, Hong J-M, Teran J, Fedkiw R. Fracturing rigid materials. *IEEE Transactions on Visualization and Computer Graphics* 2007; **13**(2): 370–378.
9. Müller M, McMillan L, Dorsey J, Jagnow R. Real-time simulation of deformation and fracture of stiff materials, In *Proceedings of the Eurographic Workshop on Computer Animation and Simulation*, Manchester, UK, 2001; 113–124.
10. Parker EG, O'Brien JF. Real-time deformation and fracture in a game environment, In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, New Orleans, Louisiana, USA, 2009; 165–175.
11. Pauly M, Keiser R, Adams B, Dutré P, Gross M, Guibas LJ. Meshless animation of fracturing

solids. *ACM Transactions on Graphics* 2005; **24**(3): 957–964.
12. Liu N, He X, Li S, Wang G. Meshless simulation of brittle fracture. *Computer Animation and Virtual Worlds* 2011; **22**(2–3): 115–124.
13. Müller M, Teschner M, Gross M. Physically based simulation of objects represented by surface meshes, In *Proceedings of the Computer Graphics International*, Hersonissos, Crete, Greece, 2004; 26–33.
14. Molino N, Bao Z, Fedkiw R. A virtual node algorithm for changing mesh topology during simulation. *ACM Transactions on Graphics* 2004; **23**(3): 385–392.
15. Steinemann D, Otaduy MA, Gross M. Splitting meshless deforming objects with explicit surface tracking. *Graphical Models* 2009; **71**(6): 209–220.
16. Liu MB, Liu GR. Smoothed particle hydrodynamics (sph): an overview and recent developments. *Archives of Computational Methods in Engineering* 2010; **17**(1): 25–76.
17. Paul WC, Rajarshi D. The potential for sph modelling of solid deformation and fracture, In *IUTAM Symposium on Theoretical, Computational and Modelling Aspects of Inelastic Media*, Vol. 11, Cape Town, South Africa, 2008; 287–296.
18. Lucy LB. A numerical approach to the testing of the fission hypothesis. *Astronomical Journal* 1977; **82**: 1013–1024.
19. Gingold RA, Monaghan JJ. Smoothed particle hydrodynamic: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society* 1977; **181**: 375–389.
20. Müller M, Dorsey J, McMillan L, Jagnow R, Cutler B. Stable real-time deformations, In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, San Antonio, Texas, USA, 2002; 49–54.
21. Peter K, Sebastian M, Mario B, Gross M. Flexible simulation of deformable models using discontinuous galerkin fem, In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Dublin, Ireland, 2008; 105–115.
22. Becker M, Ihmsen M, Teschner M. Corotated sph for deformable solids, In *Proceedings of the Fifth Eurographics Conference on Natural Phenomena*, Munich, Germany, 2009; 27–34.
23. Tanaka M, Sakai M, Ishikawajima H, Koshizuka S. Rigid body simulation using a particle method, In *ACM SIGGRAPH 2006 Research Posters*, Boston, Massachusetts, USA, 2006. Article No. 132.
24. Rankine WJM. On the stability of loose earth. *Philosophical Transactions of the Royal Society of London* 1857; **147**: 9–27.
25. Vladimir EC. Why so many clustering algorithms: a position paper. *ACM SIGKDD Explorations Newsletter* 2002; **4**(1): 65–75.
26. Bender J. Impulse-based dynamic simulation in linear time. *Computer Animation and Virtual Worlds* 2007; **18**(4–5): 225–233.
27. Müller M, Charypar D, Gross M. Particle-based fluid simulation for interactive applications, In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, San Diego, California, USA, 2003; 154–159.

## AUTHORS' BIOGRAPHIES

**Feibin Chen** is currently a final year PhD student at the College of Civil Engineering, Tongji University, China. He received his BE degree in Civil Engineering from Fuzhou University in 2006. His research interests include physically based animation and virtual reality.

**Changbo Wang** is a professor of Software Engineering Institute, East China Normal University, China. He received his PhD degree at the State Key Lab. of CAD&CG, Zhejiang University in 2006 and received BE 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 visualization, and so on.

**Buying Xie** is a professor at the College of Civil Engineering, Tongji University, China. He received his BE degree in 1982 and ME degree in civil engineering in 1984, respectively, both from Tongji University. His research interests include computer simulation in civil engineering and virtual reality.

**Hong Qin** is a full professor of the Computer Science in Department of Computer Science at State University of New York at Stony Brook (Stony Brook University). He received his BS (1986) degree and his MS degree (1989) in Computer Science from Peking University in Beijing, China. He received his PhD (1995) degree in Computer Science from the University of Toronto. His research interests include Computer Graphics, Geometric and Physics-based Modeling, Computer Aided Design, Computer Aided Geometric Design, Computer Animation and Simulation, Virtual Environments and Virtual Engineering, and so on.


*Comp. Anim. Virtual Worlds* 2013; **24**:215–224 © 2013 John Wiley & Sons, Ltd.
DOI: 10.1002/cav