Advances in Computer Science Research, (ACSR), volume 52
2016 International Conference on Computer Engineering and Information Systems (CEIS-16)

Physical Solid Fracture Simulation Based on Random Voronoi Tessallation

Shao-Xiong Zhang, Chen-Yang Chen,
Zhang-Ye Wang
State Key Lab. of CAD&CG,
Zhejiang University,
Hangzhou 310058, P. R. China
E-mail: zywang@cad.zju.edu.cn

Fan Zhang
College of Computer Science and Technology,
Zhejiang University of Technology,
Hangzhou 310014, P. R. China
E-mail: zhangf@ zjut.edu.cn

***Abstract*-Physics based solid fracture simulation techniques have great applications in the domains such as Virtual Reality, Military Affairs training, special effect manufacture in films & television, and game design, etc. However, current solid fracture simulation algorithms have the defects of either too slow or unrealistic. To overcome this shortcoming, in this paper we present a novel algorithm to real-time simulate the fracture process of solid with quite realism. To accelerate the fracture process, we hire random Voronoi Tessallation which can significantly decrease the number of crack meshes. To make the fracture process more realistic, we adopt a Physics based method called symmetric Galerkin BEM(Boundary Element Method) to stress analysis, and thus generate more detailed fracture results. We also devise a efficient crack propagation algorithm to simulate the fracture process. Experiments result show that our solid fracture algorithm can realistic simulate various solid fracture processes in real time.**

Keywords-real-time solid fracture process; Physics based simulation; random voronoi tessellation; boundary element method

I. INTRODUCTION

Techniques for simulating brittle fracture have been widely used in different areas like film & television industry, games and virtual reality. Physically-based approaches which simulate the generation and propagation of cracks can produce highly realistic results. For instance, O'Brien et al[1] simulates the cracks through a finite element method. However, these methods always would take long time of computation.

The approach which computes the locations of fragment centers through Voronoi fracture algorithms offers another solution. This method could fast simulate the fracture of brittle objects. But, fracture simulations over structured grids and Centroidal Voronoi Tessellations CVTs produce unrealistic cracks. Bolander et al[2] generate cracks initiate and propagate more realistically through a random method.

We introduce a novel Voronoi fracture method. Based on the random Voronoi cells which have been calculated in advance, we analyze the stress tensors computed over a boundary element method. Then we can determines where cracks should initiate and in what directions they should propagate.

Our main contributions are listed below:

- We propose a random Voronoi meshing algorithm to generate meshes over 3D domain with prescribed boundaries which greatly enhance the simulation efficiency and make the real-time solid fracture process possible.
- To make the fracture process more realistic, we adopt a Physics based symmetric Galerkin boundary element method to generate realistic solid fracture result.
- We design a a efficient crack propagation algorithm to further accelerate the solid fracture simulation and make the process more smooth.

II. RELATED WORK

A. Voronoi Tessallation

Voronoi Tessallation is a subdivision of the space into n cells, each cell corresponding to all points of the segment area. Voronoi Tessallation have multiple applications in computer science, chemistry, etc. In this paper, we just use Voronoi Tessallation to transfer the rigid object to a Voronoi Diagram, each segment represents an area of points, so that the efficiency of physics based simulation could be significantly improved.

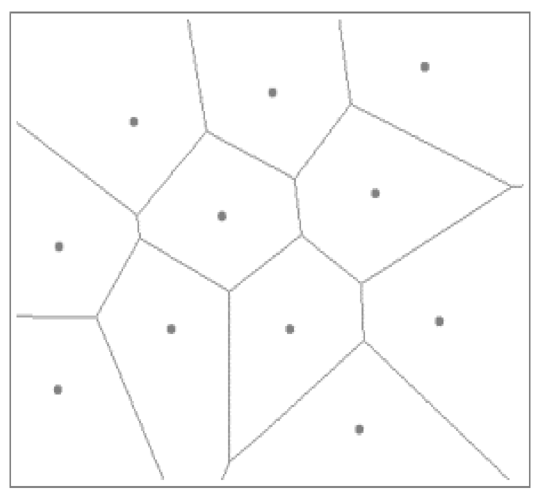

Figure 1. Voronoi tessallation diagam.

One algorithm to generate Voronoi Tessallation was invented by Steven Fortune in 1986[3]. Fortune's The algorithm maintains both a sweep line and a beach line, which both move through the plane as the algorithm progresses. The algorithm maintains as data structures a binary search tree describing the combinatorial structure of the beach line, and a priority queue listing potential future events that could change the beach line structure. The total time complexity is O(nlogn).

Another classic method of Voronoi Tessallation was proposed in Sara Schvartzman in 2014[4], named Centroidal Voronoi Tessallation(CVT). CVT is a special type of Voronoi tessellation or Voronoi diagrams. A Voronoi tessellation is called centroidal when the generating point of each Voronoi cell is also its mean (center of mass). It can be viewed as an optimal partition corresponding to an optimal distribution of generators. A number of algorithms can be used to generate centroidal Voronoi tessellations, including Lloyd's algorithm for K-means clustering.

B. *Physics Based Fracture Simulation*

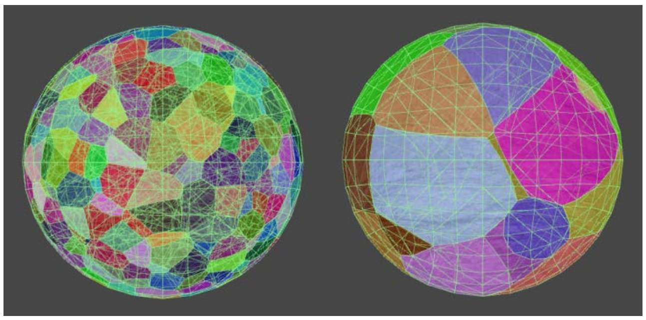

Figure 2. Generating voronoi mesh with different limitation.

To ensure realistic simulation, we devised a mathematics model based on physics. Yufeng Zhu in 2015 present a new brittle fracture simulation method based on a boundary integral formulation of elasticity and explicit surface mesh evolution algorithms[5]. In this paper, the tracking progress of a fracture propagation contains steps including mesh-based surface tracking, contact force model and fracture criteria, fracture initiation and propagation and finally remeshing operation. It's accurate and computation economically, but this method rely on surface mesh, the final result could be further improved by using other physics model, like boundary element method.

David Hahn present a method for simulating brittle fracture under the assumptions of quasi-static linear elastic fracture mechanics (LEFM). Using the boundary element method (BEM) and Lagrangian crack- fronts, and produce highly detailed fracture surfaces. Starting from a given surface mesh, convert it to an implicit surface and construct a coarse triangle mesh for the BEM. Then apply given boundary conditions and compute surface displacements and stresses on the coarse mesh. After stresses analysis, simulate crack initiation and propagation using devised algorithm. Finally, find disconnected geometry components and convert them into high-resolution meshes. This method is capable of producing highly detailed results, but the time complexity is relatively high.

In our method, we using a random Voronoi Tessallation to get high efficiency and BEM to enhance the realism of fracture simulation.

III. SOLVER

A. *Voronoi Meshing Method*

Many Voronoi meshing algorithms are based on Centroidal Voronoi Tessellations. And the seed is at the center of mass of its cell. In order to get the seed location, Du et al [6] uses an iterative method to adjust the position each time.

While it is possible to generate well shaped cells using CVTs, the geometric regularity of seed locations that arises is particularly undesirable for the fracture simulations.

We adopt a Maximal Poisson-disk Sampling method which has also been mentioned by Mohamed[7] to determine the location of the seed of each Voronoi cell. Specifically, we select a set of sample points { S } from a domain D with a limitation, d: 1. the distance between each two sample points should be more than d; 2. every point x from D is within d of a sample point s_i; 3. D_i is the sub region of D outside the d-disks of the first i samples, the probability P of selecting a point from a sub region A is relate to its area:

$$\forall\, s_i, s_j \in S, i \neq j : \left\| s_i - s_j \right\| \geq d \tag{1}$$

$$\forall x \in D, \exists s_i \in S: \left\| x - s_i \right\| \leq d \tag{2}$$

$$\forall A \subset D_i,\ s_{i+1} \in S: P(s_{i+1}) = \frac{\mathrm{Area(A)}}{\mathrm{Area}(D_i)} \tag{3}$$

Each sample seed s_i can determine the shape of the Voronoi cells, Vi:

$$V_i = \{x\} \in D: \forall j \neq i, \left\| x - s_i \right\| \leq \left\| x - s_j \right\| \tag{4}$$

The Voronoi cell for any seed is constant size, and may be computed in constant time. When the distance between the points are less than 10^{-4} r, these points are treated as one. Thus, we can avoid the edges and angels are too small.

B. *Stress Analysis based on Voronoi Diagram*

The traditional fracture simulation generally first need to divide three-dimension object to get space segmentation, the most used method including points cloud and voxel segmentation. In this paper, the voronoi meshes are extracted on the basis of extracting the key characterization point cloud, and the stress analysis is carried out on the basis of meshing.

In this paper, each element on a voronoi grid is treated as a boundary element, convert fracture simulation into a linear elasticity problem. Therefore we could use symmetric Galerkin boundary element method to solve it[8]. For the case of displacement is known, Equation (5) can be used to calculate stress.

$$M_\tau(x) = T_x \int_\tau q_\tau(y) U(y,x)\, d\mu_y - T_x \int_\tau u_\tau(y) \left(t_y U\right)^T d\mu_y \tag{5}$$

Here $M_\tau(x)$ are boundary displacement. $\tau = \partial\Omega$ denotes the boundary of the computational domain. q_τ are boundary tractions, such as forces per surface area. T donates a generalized normal derivative.

For the case of tractions are known, we can use Equation (6):

$$q_\tau(x)=W_x \int_\tau q_\tau(y)U(y,x)\,d\mu_y - W_x \int_\tau u_\tau(y)\left(t_y U\right)^T d\mu_y \quad (6)$$

The meaning of each variable is same as Equation (5).

The following linear system is obtained by integrating all the conditions of each primitive, see Equation (7) below:

$$\begin{bmatrix} B & -P \\ P^T & R \end{bmatrix}\begin{bmatrix} m \\ n \end{bmatrix} = \begin{bmatrix} f_r \\ f_n \end{bmatrix} \quad (7)$$

Here m and n are unknown boundary displacements and tractions respectively, while the right hand side is assembled using the known Dirichlet and Neumann boundary data. The first row and second row represent the discretized versions of Equation (7). In the implementation, we just using he HyENA library [9] to solve the linear system above.

C. *Crack Initiation*

According to the results of the stress analysis, all the elements that exceed the threshold are filtered out according to the crushing condition criterion. We use Rankine condition in this paper, thus, we can control the volume loss that may occur during the crushing process. A new source of breakage is generated when the force of a previously unbroken element exceeds the local material strength and is less than the number of user-defined broken blocks.

The specific steps for our fragmentation algorithm are as follows:

```
Algorithm initiation
1: clear the FRACTURE BUFFER, FB
2: according to the stress, sort D into descending
order
3: for each node v in D
4:      if v has already been found in FB
5:           continue
6:      else
7:           crack = Crack node(v)
8:           add crack to FB
9: end
```

D. *Crack Propagation*

According to the gradient of the broken source and the surrounding forces, we can get a vector of broken direction. In order to obtain a broken crack, we use the following formula in the material space to calculate the speed of each active node.

$$B_{cp}=a\left(u-p_e(u)\right) \quad (8)$$

Here u is the unit vector bisects the angle composed of the corresponding crack node and its two nearby crack nodes, p_e is the orthogonal projection onto the principal eigenvector e of stress tensor.

If a vertex in the primitive is marked as an active node, the break expansion operation is applied. New active nodes and edges may be generated during the expansion, and newly generated edges and nodes may be used as new front ends.

For all active nodes, the crack propagation algorithm is used to split or merge the cracked cracks. Crack propagation algorithm is as follows:

```
Algorithm propagation
1: dt = time_step
2: for eacj crack c in FB
3:      Calculate Node position(c, dt)
4:      for each active node v in c
5:           if v at the boundary
6:                set v.active = false
7:           end if
8:      end for
9:      meshing (c)
10:     smoothing (c)
11:     Preserve Feature (c)
12:     Crack Extension (c)
13:     if no active node
14:          Update Mesh
15:          Remove c
16:     end if
17:end for
```

E. *Grid Adjustment and New Substance Generation*

In this paper, the grid update method is mainly applied to the edge of the segmentation, flip and so on. Specific processes include: First, When the grid adjustment algorithm to get a separate grid structure and grid representation of the physical entity size is greater than the user-set threshold, generate a new object. Second, The active edges of all mesh edges are reinitialized to the inactive edges, thus supporting multi-stage fragmentation simulations. Third, in each update stage, the cracked crack is smoothed, making it closer to the physical real effect.

IV. RESULTS

We have applied our fracture algorithm in the Unity 5.3.0. The model is treated as rigid body and the Random Voronoi mesh is calculated. By adding force on the model, the result of fracture simulating would be got.

Figure 3. Fracture process of a wall.

Figure 4. Fracture process of a stone ball.

We have simulate several cases on a 3.60 GHz Intel i7-4790 CPU with 8GB of RAM and NVIDIA GTX960 GPU. Figure 3 shows the simulation result of a ball crashing to the wall.

TABLE I. EFFICENCY OF BREAKING WALL

FPS: 60		
	before	***after***
Triangles number	14.5k	232.8k

TABLE II EFFICENCY OF BREAKING STONE

FPS: 60		
	before	***after***
Triangles number	6.7k	525.4k

Figure 4 shows the fracture process of a stone ball. The number of triangles is 6.7k. While the cracks are generating and propagating, the number increases and reach 525.4k. But the efficiency of the simulation has not been impacted, the rendering speed keeps stably at 60fps.

More solid fracture cases, please see Figure 5 and Figure 6. And the attached video file (SolidFractureDemo.mp4)will show the detailed fracture process of different kinds of solid objects.

V. CONCLUSION

In this paper, we proposed a novel method of fracture simulation. Firstly, we utilize a random Voronoi Tessallation to significantly improve the efficiency of simulation. Each fragment is treated as a boundary element, contribute to corresponding to stress physics model. Secondly, we use advanced BEM to enhance realism, the progress of fracture including stress analysis, crack initiation, crack propagation, mesh adjustment and new object generation. Experiment results show that our method could get a high-resolution scenery and maintain a high frame frequency at the same time.

However, our method could have further optimization when utilize multi-scale Voronoi Tessallation, each fragment is self-adaptive divided according to the weight of distribution factor. Besides that, the simulation model could be better if we could handle some extra cases such as interior explosion caused fracture and so on.

ACKNOWLEDGMENT

This paper was supported partially by 863 project of China under granted No. 2015AA016404, National natural Science funds under granted No. 61272302 and 61303141.

REFERENCES

[1] James F. O’Brien Jessica K. Hodgins. “Graphical Modeling and Animation of Brittle Fracture.,” Conference on Computer Graphics & Interactive Techniques, 1999, pp 137–146.

[2] J. E. Bolander Jr. and S. Saito. “Fracture analyses using spring networks with random geometry, ” Engineering Fracture Mechanics, 1998, 61:569–591,.

[3] S Fortune, Murray Hill, New Jersey. “A sweepline algorithm for Voronoi diagrams,” Proceeding SCG '86 Proceedings of the second annual symposium on Computational geometry, 1987, pp. 313-322

[4] Y Zhu,R Bridson,C Greif. “Simulating rigid body fracture with surface meshes,” Acm Transactions on Graphics, 2015, 34(4):1-11

[5] D Hahn,C Wojtan, “High-resolution brittle fracture simulation with boundary elements,” Acm Transactions on Graphics, 2015, 34(4):1-12

[6] Qiang Du, Vance Faber, and Max Gunzburger. “Centroidal Voronoi tessellations: Applications and algorithms,” SIAM Review, December 1999, 41(4):637–676.

[7] MS Ebeida,SA Mitchell, “Uniform Random Voronoi Meshes”, Springer Berlin Heidelberg, 2011: 258--275

[8] David Hahn, Chris Wojtan, “Fast approximations for boundary element based brittle fracture simulation,” Acm Transactions on Graphics, 2016, 35(4):1-11

[9] HYENA, 2013, Hyperbolic and Elliptic Bymerical Analysis. Graz University of Technology. http://www.mech.tugraz.at/HyENA v. 2013-Nov-04

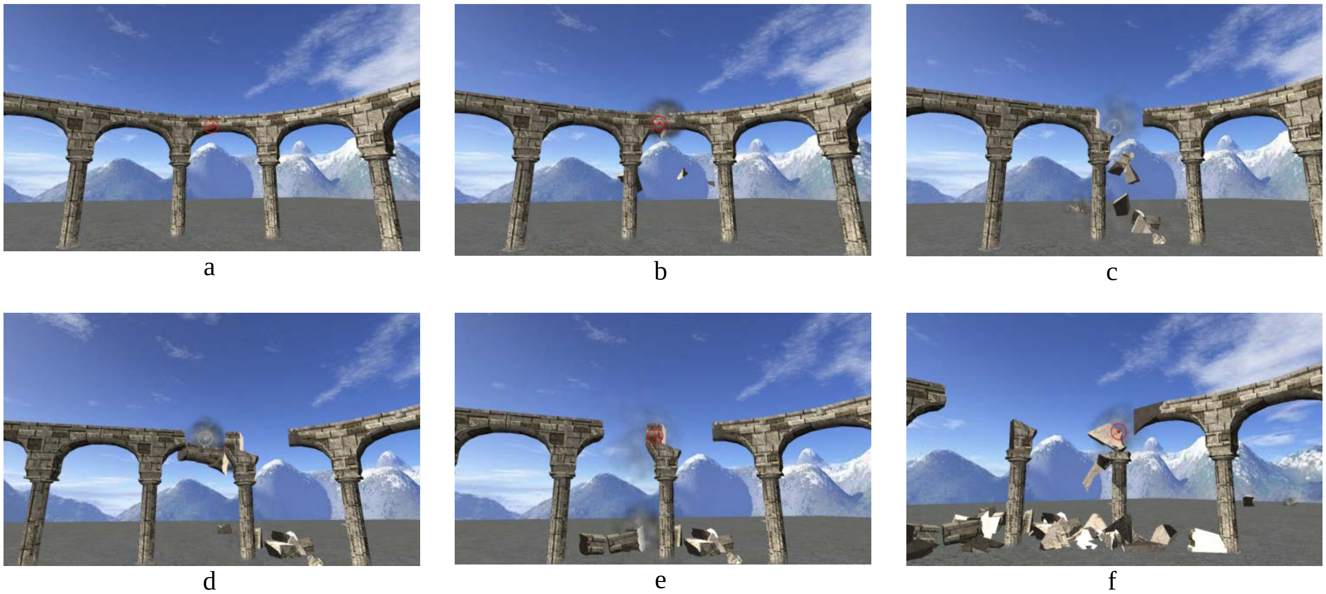

Figure 5. The Fracture process of a pillar shot by bullets.

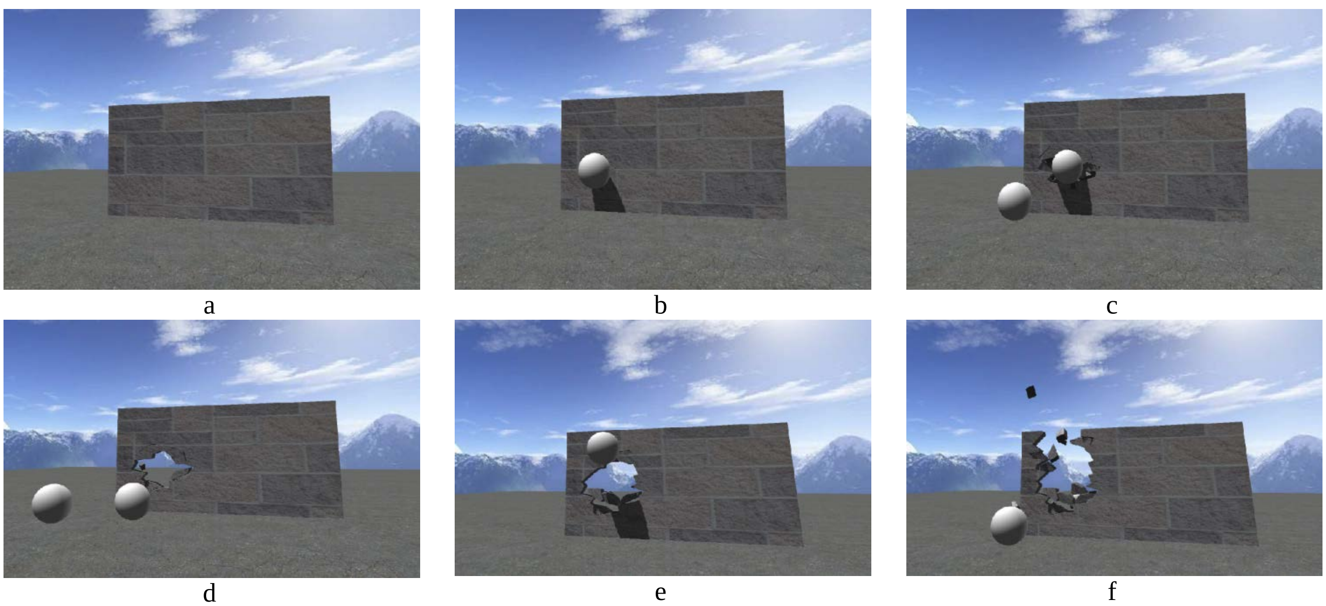

Figure 6. The Fracture process of a stone wall hit by balls.