

## Crack pattern simulation based on 3D surface cellular automata

Stéphane Gobron,  
Norishige Chiba

Iwate University, 020-8551 Morioka, Ueda 4-3-5,  
Japan  
E-mail: stephane@cis.iwate-u.ac.jp

This article describes a method for modeling the propagation of cracks on any 3D surface. This method allows almost any type of cracks on any type of triangulated 3D object. Our model's main advantage is that it proposes a semi-physical solution, making it both user controllable and easily extensible. We first introduce the general development of cracks. We then present our original model of spectrum stress, followed by a description of the mutual interaction between cracks and stresses. Then, we describe special rendering techniques including the multi-thickness anti-aliasing linked-segment method and the crack mirror special effect. The final section presents intermediate graphical results that review the entire model as well as a set of different crack patterns using various types of material such as concrete, ceramic, mud, and glaze.

**Key words:** Cellular automaton – Cracking – Hyper-texture – Multi-layer modeling – Simulation

## 1 Introduction

Nowadays one of the main topics in the field of Computer Graphics (CG) is the realization of realistic texturing on the surface of 3D objects. This article describes a method for modeling the propagation of many types of cracks on any 3D surface. In this section, we first summarize the relevant literature, then discuss the improvements which form the basis for our model, and how to achieve these improvements.

The background of this research can generally be subdivided into two main classes. Physical approaches propose realistic models but are largely restricted by huge computational times and (often) poor resulting images. We cite only a few of the best articles in this field: modeling inelastic deformation (Terzopoulos and Fleischer 1988), simulation of 3D cracks (Hirota et al. 2000), generation of crack patterns with a physical model (Hirota et al. 1998), experimental study of mud crack patterns (Neff and Fiume 1999), animation of fracture by physical modeling (Norton et al. 1991), study of fracture in microsphere monolayers by experiment and computer simulation (Skjeltorp and Meakin 1988), and recently in SIGGRAPH'99, graphical modeling and animation of brittle fracture (O'Brien and Hodgins 1999). The main differences between our work and this remarkable article (O'Brien and Hodgins 1999) are as follows: An object's fracture in 3D is certainly much more impressive than the propagation of cracks on 3D surfaces, but it doesn't serve the same purposes. In particular, we show that in our model there is a wide range of materials and many generated crack patterns; that cracks start, evaluate, interact, and stop over time; that their velocities and curvature depend on their history; and especially that no external force is needed to generate them.

Models in the second class, such as texture mapping techniques, are often called "faking models" (see Barzel (1997) for an excellent explanation of the need for "fakes" in CG). For example, one of the most recent publications on "ultra-realistic" texturing using texture mapping proposes a general approach based on reflectance and texture (Dana et al. 1999). These models are very convenient, since they are easy to implement, quick to compute, and give impressive results. However, mapping techniques also have many limitations since they require large texturing libraries, mapping orientation is often difficult, scaling and resolution

problems arise, texturing continuities at edges are usually inconsistent, and real 3D texturing extension is almost impossible. Still, faking approaches are simple and efficient models for simulating continuous cracks using slopes attraction as shown by Chiba et al. (1991).

CG requires automatic textural effects that do not have these limitations and are intuitive enough to be implemented and computed. Unfortunately, the literature in this particular area is very limited. Models showing interesting results for automatic texturing simulation, e.g. metallic patina (Dorsey and Hanrahan 1996), dust accumulation (Hsu and Wong 1995), surface imperfection (Wong et al. 1997) – have been published only since about 1994. Recently, we proposed a more general approach introduced for 2.5D in Gobron and Chiba (1997) and detailed in 3D in Gobron and Chiba (1999) using 3D surface cellular automata (CA) (see Chaudhuri et al. (1997), Codd (1968), and Norton et al. (1991) for CA theories). CA can be very useful in visual simulation because we can identify the object not only as a set of polygons covering the visible surface, but also as a set of material layers subdivided into regular cells with independent behaviors. Furthermore, our model can be described as a ‘meta-Hypertexture’ – see Perlin (1989) for Hypertexture definition – as it permits non-null multi layers to be defined with a potential of mutual interactions. This allows the object to become dynamic: it becomes alive through time.

Articles or books referring to CA in CG are relatively easy to find, e.g., Fleischer et al. (1995), Fowler et al. (1992), Rosenfeld (1979), Stanley and Ostrowsky (1986), Sternberg (1980), Takai et al. (1995), Turk (1991), but those dealing specifically with surface CA for generating automatic texturing are fairly rare (Turk 1991). Thus, this field remains largely open for new discoveries.

For more explanation and classification of most of the articles previously cited, please refer to Gobron and Chiba (1999).

Based on our 3D CA model, we present a method for simulating realistic propagation of various types of cracks. The crack propagation is automatically generated using an original “intuitive-physical” approach. It is *physical* because from a CG scientist’s point of view this model can appear quite theoretical, but in reality our model is far from realistic from a physicist’s point of view, of course – and therefore remains *intuitive*.

## 1.1 Overview

This paper is organized as follows.

Section 2 gives an overview of cracks and stress, explaining what cracks are and why they appear. Section 3 defines our stress model, i.e., the causes of cracks. In Sect. 3.1, we first propose an intuitive notion, called *spectral* stress, for simulating the stress occurring at each cell, and then present a solution method for the average spectrum showing its pre-computation, introducing a method for simulating the material elasticity (see Sect. 3.1.2). Section 3.2 deals with the average stress spectrum. Section 4 discusses crack generation and propagation. We first show, in Sect. 4.1, how to determine the orientation and propagation of *crack modules* (CMs) and then show – in Sect. 4.2 – how CMs release the stress field surrounding their path in our model. The rendering, in Sect. 5, is presented as an extension of a previous work on surface CA. Section 5.1 presents a technique for a better visual representation of the crack’s patterns (using an anti-aliased segment). Section 5.2 proposes a solution to the transparent layer problem where shadows and light effects appear. Section 6 verifies the advantages of this model with a description of pre-computation results and resulting images. We conclude the paper in Sect. 7 with a brief description of the potential expansion of this work.

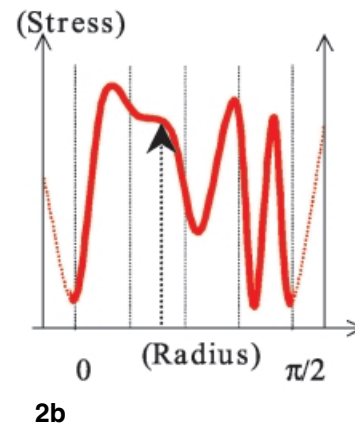
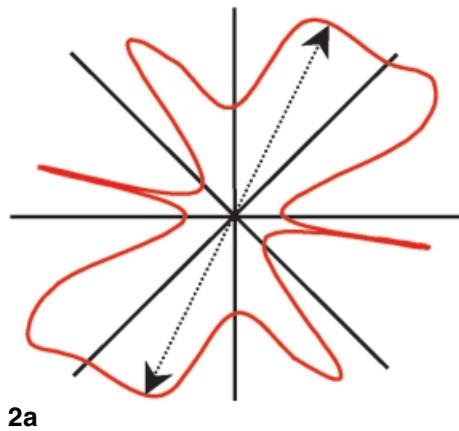
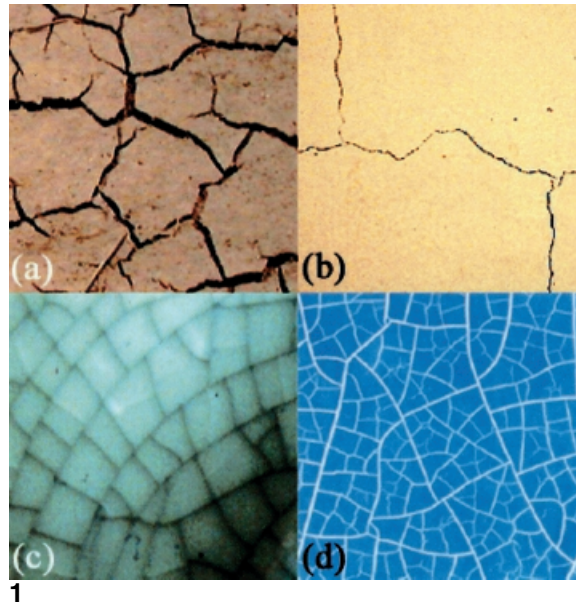
## 2 General concepts of stresses and cracks

We begin by reviewing some simple and general considerations: What are cracks? Why do cracks happen? What are stresses or stress fields? How can they be represented? What is one of the most important achievements of this paper?

A crack is the systematic breaking of material liaisons (connections) through a continuous but sometimes non-derivable line. Its shape varies depending on the material (see Fig. 1), the object’s geometry, and possibly outside constraints.

A crack appears when the internal stresses (tensions) of a material are greater than the material resistance: A solid liaison first breaks under the stress, then the neighbor, and so on, as springs would in a domino fashion.

What then is a *stress* and what is a *stress field*? Stresses arise as the material deforms. We define a stress as the multi-directional tensions that exist

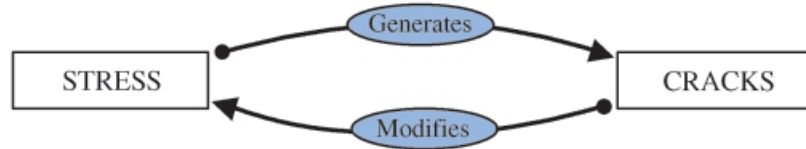


**Fig. 1a–d.** Example of real crack patterns: **a** mud; **b** concrete; **c** glaze (first layer) and ceramic (second layer); **d** plastic painting; **Fig. 2a,b.** Stress representation per surface area: **a** example of tension (stress) at a point and **b** corresponding spectrum

in a material per infinitely small unit size; it can be a compressive stress or a tensile stress or both depending on the direction. A stress field is the set of stresses over an entire region, surface, or 3D object (note that it is not the sum).

It is difficult to represent a stress – especially in 3D – since it can go in any direction with various intensities. However, we are working on 3D surface layers, which gives us an excellent advantage. Since the surface is much larger than the thickness of the layer, we assume that the dimension orthogonal to

the surface can be ignored. Experiments tend to show (but not prove) that the thickness of the layers should not greatly exceed the cell size dimension in order for the crack initialization, orientation, and velocity predictability to remain valid. However, this problem of resolution is solved by the local cell subdivision described in the crack model description. Figure 2a is a possible intuitive representation of a stress in a very small region of material. Note that the tension in one direction must be equal to that in the opposite direction (see black double arrow). Therefore



**Fig. 3.** The relationship between stress and cracks

the graph has a central symmetry, and is represented Fig. 2b. This theoretical stress spectrum is simplified as shown in Sect. 3.

A strong relationship exists between stress and cracks. We think that understanding this relationship is the key to generating (and even simulating or predicting) any type of cracks on any type of material. This paper does not pretend to solve this very complicated physics problem, but presents an initial attempt to simulate very roughly and intuitively the recursive function between crack and stress (as shown in Fig. 3).

For further explanations of the physics of fractures, please refer to Broek (1991) or Davidge (1979).

### 3 Stress model

Our model for generating crack patterns is based on a “semi-realistic” crack behavior approach. We consider it “semi-realistic” because it uses simple physical and material properties, keeping in mind the limitations of the computer. The domain where the cracks evolve is a 3D surface CA, described in Gobron and Chiba (1999), with the important restriction that we do not consider the stress interaction between layers. The crack simulation can therefore be applied to any 3D object that can be simulated with layers, i.e., the input object has an inside volume with which this crack model does not interact, and its layers are non-null. Most objects satisfy this definition (e.g., any revolution object, most ceramics – e.g. extra glaze layer –, and metallic objects covered by paint or plastic), but applying our model to, for instance, a piece of rock is not possible because it is not possible to define at least one layer.

We logically begin the description of our model with the presentation of stress, since it is the cause of the

cracks. The following section details our model for simulating the stress that occurs on the 3D surface layers.

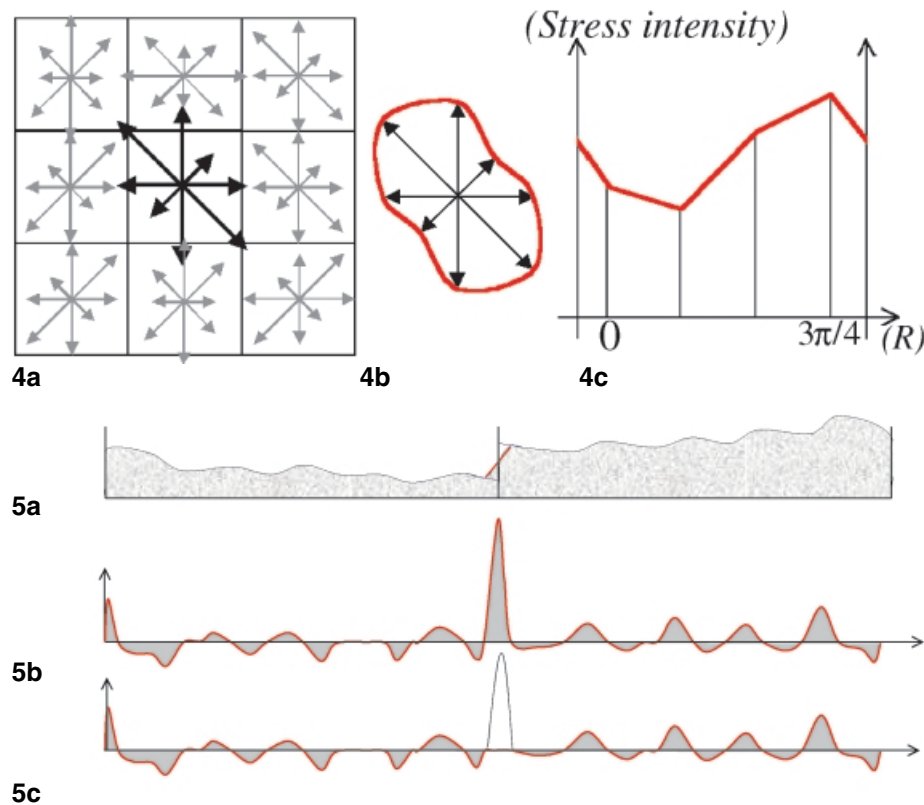
After briefly showing the reasoning used to derive the concept of a *Stress Spectrum*, we describe our stress model structure in Sect. 3.1: how it is pre-computed and why it is needed, and the relation between material elasticity and the stress field. Then, in Sect. 3.2, we present a method for the generation of cracks by stress.

Our first approach to approximate the stress was to define it as a single 2D vector. This method seemed to be very convenient, since it was easy to implement and gave the main direction of the stress as well as its intensity. However, by definition, a vector cannot represent multiple directions; this model was therefore too simple to adequately represent the stress. We then studied what we call an “elliptic stress representation” which consisted of two orthogonal vectors. However, this model makes it difficult to find the stress linear average (see Sect. 3.2), and it provides too rough an approximation of the real tension orientations. Keeping in mind that the stress representation should both follow the CA grid – for low computational cost – and allow multiple stress directions, we found a convenient model that we call the *stress spectrum*, which is described below.

#### 3.1 Stress spectrum

As mentioned, this *stress spectrum* model is set on a 3D surface CA and we assume that a stress field is defined for each cell of the surface layer. To minimize the amount of memory and optimize the stress intensity computation, and since the surface of each cell is a square, the eight Cartesian directions define the stress spectrum (see Fig. 4a and b).

In the following subsections, we first describe how to compute the stress spectrum and discuss an intensity



**Fig. 4a–c.** Stress Spectrum graphical representation: **a** one cell, one stress; **b** stress field; **c** stress spectrum  
**Fig. 5.** Discontinuity surface problem generating gaps in stress intensity

problem concerning surface discontinuity, and then present a simple and efficient method for simulating the material elasticity property.

### 3.1.1 Stress spectrum intensities computation

In Nature, most cracks do not appear because of shock or collision. For example, ceramics generate their typical crack patterns because of large differences in temperature; both the clay and the painting will do so, due to the drying process. Thus external forces are not taken into account directly, and the stress is computed depending on the surface geometry and the material.

We assume that each of the stress directional intensities is proportional to the difference of opposite cell layer thicknesses, and to the layer curvature in this direction. We immediately notice that for such a spectrum model only four intensities are required, corresponding to the following angles from

zero to  $\pi$ :

$$(0), \left(\frac{\pi}{4}\right), \left(\frac{\pi}{2}\right), \left(\frac{3\pi}{4}\right)$$

A special case occurs when it is not directly possible to find the cell's neighbor; this happens when cells at the border of the object polygons are not regularly aligned. In this case, we find the nearest cell on the neighboring facet by comparing the distance between their respective centers (see Gobron and Chiba (1999) for details of the facet-to-facet cell access model). An exception occurs when the angle between the neighbor's layer and the current cell's layer is approximately zero. Since our previous CA model was restricted to triangles, the complex facet polygons of any object must be subdivided into sets of triangles. Then even if the same surface noise is assigned to each triangle of the same layer, discontinuities arise at the edges.

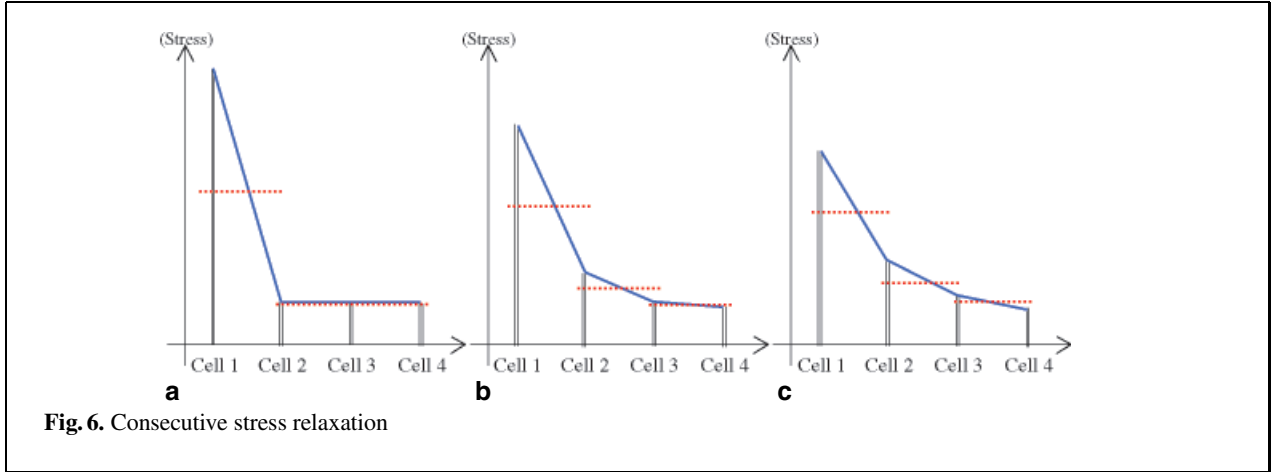


Figure 5 illustrates this problem and the corresponding solution. Note that this fix solves the stress high-intensity discontinuity problem that occurs fairly rarely (at most once per thousands of cells considering a minimum level of triangle to cells subdivision). Schema (a) presents a side view of two neighboring layers with parallel facets; we can see that the noise of the thickness is the same, but to maintain continuity between the two layers, their surfaces have been forced to match (using the average thickness). Schema (b) shows the derivative of the thickness (which is equivalent to the stress intensity in our model); we can observe in the center an incorrect “jump” in intensity. Schema (c) shows the corrected stress computation: We set to null the stress spectrum at the edge of same plane facets. This technique permits us to artificially eliminate sudden gaps of stress intensity that could generate unnatural crack patterns, but also creates unnatural null-stress spectrums at some edge regions. Fortunately, this drawback is diminished by the material elasticity described in the following subsection.

### 3.1.2 Stress relaxation

After the stress pre-computation for all cells, very large differences of spectrum intensity, as well as artificial null-stress spectrum regions, appear (due to the discontinuity in geometric texture from facet to facet). In Nature, every material has an elasticity coefficient that permits it to share tension at any point with its neighboring structure. This coefficient varies according to the material, and is almost null for some ceramics, but very large for materials such as rubber. We simulate this phenomenon in our model just after

the stress spectrum initial pre-computation with the following algorithm.

For each type of material, we assign two parameters: the elasticity loop  $e_L$  and the elasticity sharing percentage  $e_S$ . Then for each layer  $l$ , we iterate from 1 to  $e_{L,l}$  and for each cell  $C$  we share  $e_S$  percent of the difference with the average  $n$  neighbors that are inferior to the intensity of  $C$ . Figure 6 depicts the idea, and more information on these parameters can be found in Appendix A.

The following formula summarizes the algorithm with:

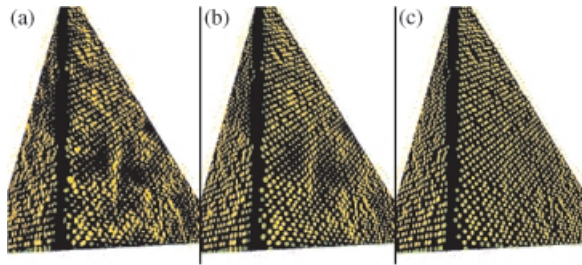
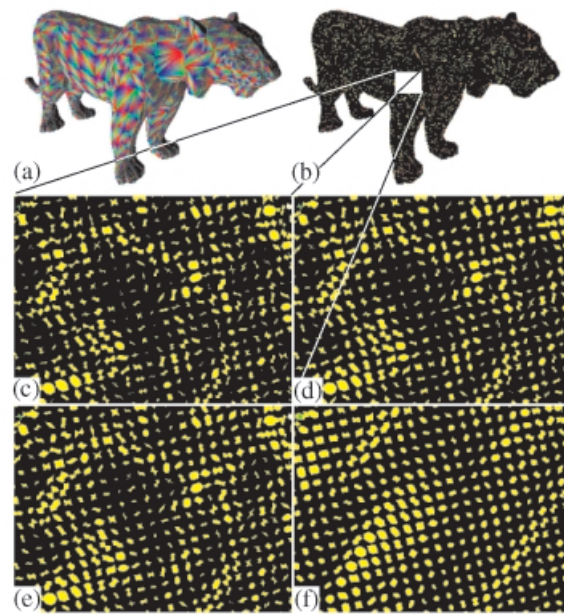
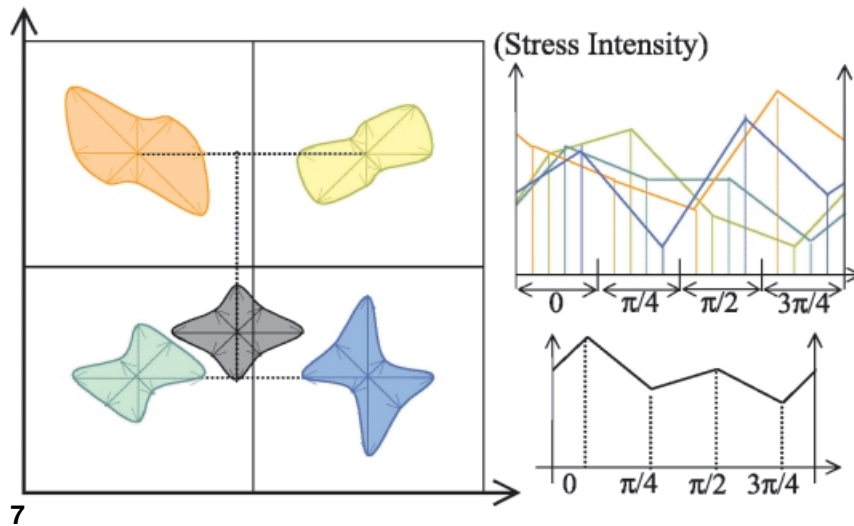
- \* Sigma being the “sharing intensity”;
- \* Current cell  $C$  at coordinates  $(x, y)$ ;
- \* Function  $\inf()$  being the test for tensor intensity inferiority compared to the current cell intensity (see Sect. 4.1 for more explanation of this function);
- \*  $(i, j)$  parameters for the neighboring cell ( $i$  and  $j$  from  $-1$  to  $1$ ).

$$\begin{aligned}
 & * k_1 : \left( \begin{matrix} x \\ y \end{matrix}, l \right) \& k_2 : \left( \begin{matrix} x+i \\ y+j \end{matrix}, l \right) \& \inf(C_{(k_2)t+1}) \\
 & = \inf(C_{(k_2)t}) + \frac{\sigma_{(k_1)t+1}}{n} \\
 & \sigma_{(k_1)t+1} = 1 - e_P \cdot \frac{\sum_{k=1}^n \inf(C_{(k_2)t})}{n}
 \end{aligned}$$

Effects over the stress field are presented in Plates 1 and 2.

Now that we have computed the stress field on the entire object surface, a final consideration has to be discussed before we start to generate cracks: the inter-cell stress spectrum.



**Plate 1****Plate 2****Plate 1.** Change in material property over a tetrahedron after 1, 7, and 100 steps with  $e_P = 30\%$ **Plate 2.** Material elasticity over a fairly complex object**Fig. 7a–c.** Computation of the average stress spectrum at any position: **a** of the surrounding four stresses; **b** corresponding stress spectrums; **c** resulting average stress spectrum

### 3.2 Average stress spectrum

As we will see in Sect. 4, cracks have to be much more precise than cells, thus to avoid sudden changes of stress intensity, we need to know the stress field not only at a cell region but also at any position be-

tween the cells. We assume that the stress field is linearly continuous which allows us to compute the average stress with a bilinear interpolation.

Figure 7 expresses graphically the result for a stress spectrum with a horizontal coefficient of 40% and a vertical coefficient of 20%. We can see that the re-

sulting stress spectrum (in black) is similar to that of the nearest cell (in green).

We have defined the stress field at any position of the 3D multi-layer CA. From this solid basis, we detail how the cracks are generated on the object surface in the following section.

## 4 Crack modules (CM)

Crack propagation is determined by the systematic stress release of all the *unstable* cells of the input object. We define *unstable cells* to be those that contain at least one stress directional intensity stronger than the material resistance  $m_R$  (see Table 1 of Appendix A). To make the crack pattern even more realistic, the size of the crack depends on the stress release.

### 4.1 Crack “birth”

The initial step consists of making a dynamic list of all the unstable cells arranged in decreasing stress-intensity order. All these cells try to make a crack, but sometimes a priority crack releases some neighboring region that is also listed as a weaker potential crack. Thus before selecting a new crack from the top of the list, we must first re-order the list. Since stress spectrums allow more than one direction, it seems impossible to re-order the potential cracks. To solve this problem, one must remember that a crack is generated from the stress spectrum direction that has the highest intensity. Thus we order the potential cracks according to the highest intensity of the corresponding stress spectrum.

Since a crack can develop in multiple directions, with every crack is associated a set of *crack modules* (CMs). A CM is one of the crack’s heads, e.g., when tearing a piece of paper only one CM is produced. In our case, a crack cannot be generated with a single CM since we are working on a surface that has no beginning or end. (We conclude that it is not directly possible to simulate the tearing of a piece of paper with our model. In fact, as stated in the introduction of Sect. 3, a filled object cannot be simulated, and surprisingly a piece of paper is indeed a filled object!) Note that the re-ordering of the potential cracks also occurs after each step of movement of the current CM since they have released some stress, and the potential crack list is modified accordingly.

Determining the initial crack direction is not trivial since stress spectrums have more than two directional intensities. Thus we assume that a crack always begins with at least two directions, as suggested by To and Miyata (1998). Four types of crack pattern are defined as follow:

- “I”, the most common, consists of two CMs that are initially propagated in the opposite direction;
- “T” consists of three CMs, with two of them similar to the “I” type, and the third orthogonal to the others (essentially a resulting pattern);
- “Y” also consists of three CMs, but their initial directions form regular angles of  $120^\circ$  (typical with dry mud);
- “X” is extremely rare and consists of four CMs with orientations that form regular angles of  $90^\circ$ .

### 4.2 CM movement

Each CM moves on its material layer, releasing the stress orthogonal to its path (Sect. 4.2). Its initial orientation follows the higher stress release. At every time-step each CM is influenced by the current surrounding stress spectrum that is computed using the method explained in Sect. 3.2. It then changes its orientation or sometimes even subdivides itself, generating a fork as shall be seen later.

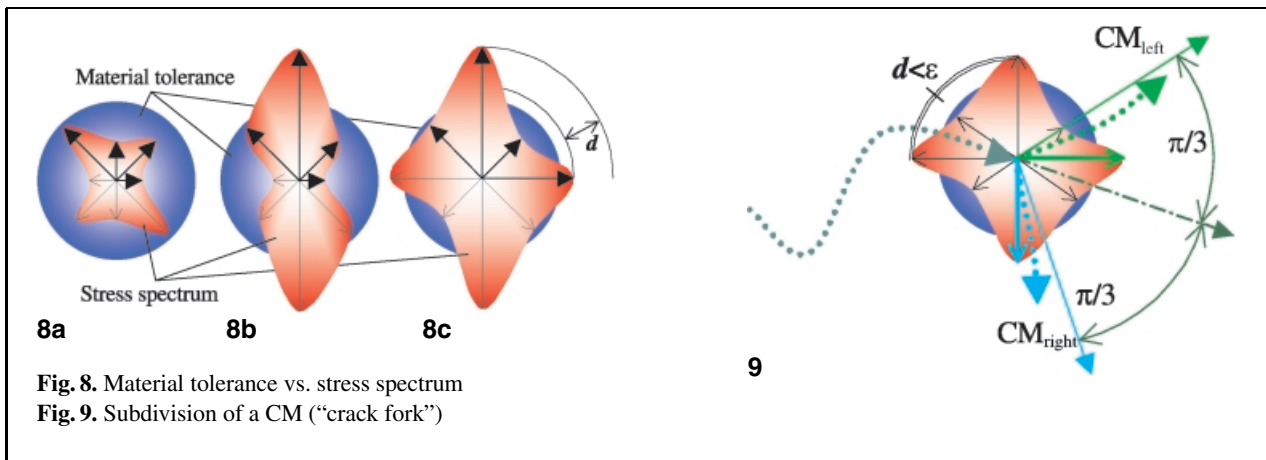
This subsection details the individual development of a CM which induces the desired crack path. We first define some general movement rules, and then explain what we call the *kinetic potential* of CMs. From this basis, we describe a CM’s initial direction and change of orientation, its movement and finally the storage of its path in the data structure of the CA.

#### 4.2.1 CM movement rules

Here are the main rules for correct development of a CM:

- CMs maneuver on the surface of the cellular object;
- CMs always stay on the same layer, but since the surface can have holes (see Hirota et al. 2000) different CMs can be propagated through different layers at the same time (this usually happens when the two layers have similar properties);
- In their movements, CMs set the encountered cells to a cracked cell status;
- CMs are terminated if:





**Fig. 8.** Material tolerance vs. stress spectrum

**Fig. 9.** Subdivision of a CM (“crack fork”)

- \* They encounter a **new** cell which already has a cracked status this rule simplifies the costly collision test problem from  $O(n \cdot \ln(n))$  to  $O(2n)$  – with  $n$  allways  $\gg 8$ ;
- \* Their kinetic potential reaches zero (see Sect. 4.2.2);
- CMs move with a step size that must be smaller than the cell size.

#### 4.2.2 CM kinetic potential

At the tip of a crack – I at CMs positions – very high and complex stress tension happens (see physic of fracture). With the current approach, this essential phenomenon is not possible to simulate directly. That is why, we attribute a fake parameter called *kinetic potential* to each CM. This property allows CMs not to stop at – and therefore be terminated – any minor obstacle. Note that this gives the curve a smoothness that is difficult to control. This is an advantage for the crack simulation of many ceramics (smooth or linear propagation), and a disadvantage for crack simulation of materials, such as mud, which require irregular cracks.

This kinetic potential is simulated by an accumulation of stress. We directly attribute a certain percentage  $p$  of the maximum stress from the encountered cell’s stress intensity. When this kinetic potential is below a certain threshold (about zero), the crack module stops.

#### 4.2.3 CM orientation

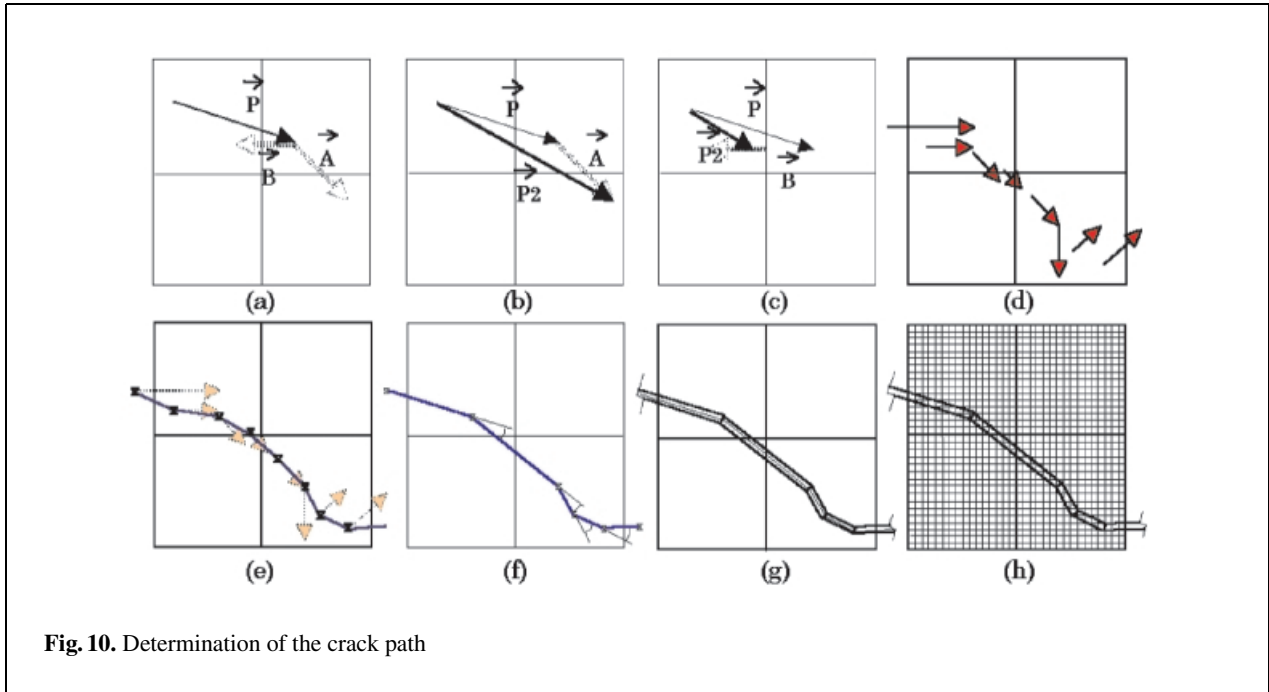
In the stress relaxation pre-computation section we introduced the layer’s elasticity property. In this subsection we present the primary material property that

makes a CM change from one orientation to another: the tolerance for stress.

All materials have a tolerance for stress, which allows a maximum tension to exist in any direction for each unit surface (in our case, each cell). If this limit is exceeded in one or more directions, it must be released so that the object becomes “stable” at this cell: cracks are generated.

Figure 8 shows the stress spectrum for three different materials: the red circle shows the resistance of each material and the red double arrows point to the local maxima. In Fig. 8a, none of the stress spectrum (red) intensity does not exceed the material tolerance (blue). In this case, the corresponding cell cannot generate a crack; CMs can go through it, but their kinetic potential decreases. In Fig. 8b, the stress exceeds the material tolerance in one direction. This cell will generate a crack if its stress intensity is not reduced the material tolerance by a neighboring crack. Figure 8c presents a case where more than one stress direction exceeds to the material tolerance. We assume that only the strongest one influences the CM direction.

A special – and very rare – case occurs when two stress directions have an intensity difference  $d$  smaller than an  $\epsilon$  value. The crack should subdivide itself into a “fork” at this position as shown in Fig. 9. The blue disk and the red regions describe respectively the material stress tolerance and the average stress at this position. In dark green, the oriented dot line represents the CM’s path before the fork, and the half-dot arrow its current orientation. The new CMs, left (light green) and right (light blue), have three



components: the light arrows show their new orientation, the bold arrow the influencing stress, and dot lines show the potential CM movement. As shown, the orientation of the new CMs always make an angle of  $\pi/3$  with the current orientation. *Note:* This crack subdivision is relatively unnecessary with the current model. Nevertheless, this property (see physics of fractures) becomes primordial when simulating cracks over non-pure material, which is probably the main future step of the current approach.

#### 4.2.4 CM paths

Figure 10 summarizes the main steps for determining our goal: the CM paths.

Schema (a), (b), and (c) show the change of kinetic potential and module orientation.

Schema (d) presents the possible set of directional influences over time due to local stress. Note that these always follow the eight Cartesian directions.

Schema (e) shows the corresponding crack path, step by step; note that the resulting path does not follow the eight Cartesian directions since the CMs are influenced by an inertia percentage ( $I_{ip}$ ).

Schema (f) presents a linked-segments solution for simplifying the crack path using a simple change of angle test ( $CS_a$ ). The smaller the angle, the better the

resulting crack pattern, but the higher the computational and memory cost.

Schema (g) shows the possible resulting crack widths along the CM path. These widths are proportional to the total stress intensity released and tend to reach the  $m_{cw}$  value (see Sect. 5, Fig. 12, and Appendix A).

Schema (h) presents a possible grid superimposed on the linked-segment. We later use this grid to compute an anti-aliasing simulation of the linked-segment. This is discussed in Sect. 5 and Appendix B.

The key to the generation of a crack is the release of the stress contained in the material, and this is described in the following subsection.

#### 4.3 Cracks modify stress

In the previous section we have shown how stress fields generate crack paths. To make our model well balanced, we propose a way to simulate the stress release surrounding CM paths.

At each sub-movement of the CM orthogonal stress only is released on both sides (hence perpendicularly to its path direction). These areas release the stress field up to a distance  $d$  with a *release intensity* dependent on the material properties, the crack thickness, and the distance between the stress-released

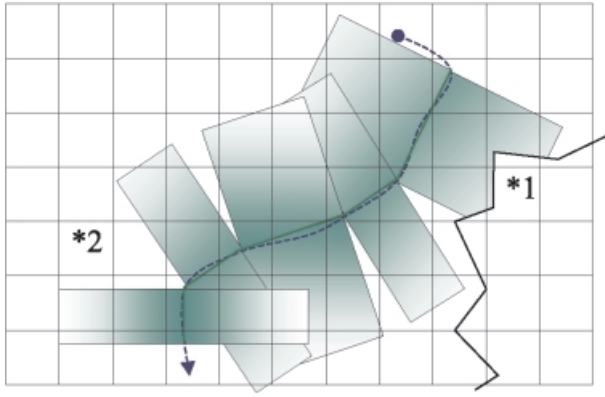


Fig. 11. Releasing stress around the crack path

region and the crack path (so that it linearly decreases). To compute this stress release, we use the anti-aliased multi-thickness linked-segment algorithm (described in Appendix B) as a mask with only one segment (the micro-crack), no cell subdivision, and a thickness of  $d$ . *Note:* The stress release range is also limited to surrounding crack's paths (see Fig. 11 '\*1'; when using our algorithm (see Appendix B, Table 2) the implementation of such phenomena is fairly easy to control by testing the loop on both sides of the crack (step 4)).

This technique is illustrated in Fig. 11, where the ideal crack path is a black oriented dot line, the CM steps are thick green segments, and the stress relaxation area (here, for purposes of simplification, its length is constant) is shown in shading from green (100%) to white (0%).

A disadvantage of this method is that occasionally if a crack head turns suddenly, some cells do not receive stress release. The cell denoted with a star '\*2' illustrates this problem in Fig. 11. Fortunately, this infrequent problem does not significantly alter the visual effect.

The interesting consequence of this algorithm is the *sequence controlled mutual interaction* between crack and stress: stress generates cracks, cracks modify the stress, stress then modifies cracks, etc.

## 5 Rendering

Cracked layers cannot be rendered using the previously described method (Gobron and Chiba 1999) based on triangle strip reconstitution since the crack path is much more precise than the cell size.

In this section, we propose a method for rendering the crack path stored as a chain segment – with independent thickness – on 3D surfaces using cellular multi-layers. The following section describes a method, taking into account complex 3D surfaces, for rendering the crack path precisely. (Note that we still use the OpenGL  $z$ -buffer (Mazarak et al. 1999) for the rendering.)

### 5.1 Anti-aliased multi-thickness linked-segment

Many aspects of anti-aliasing theories in CG have previously been discussed. Our model is specific (3D surface cellular, multi-layers, and linked-segment with multi-thicknesses, frequent data structure change), which is why we briefly describe it in this section.

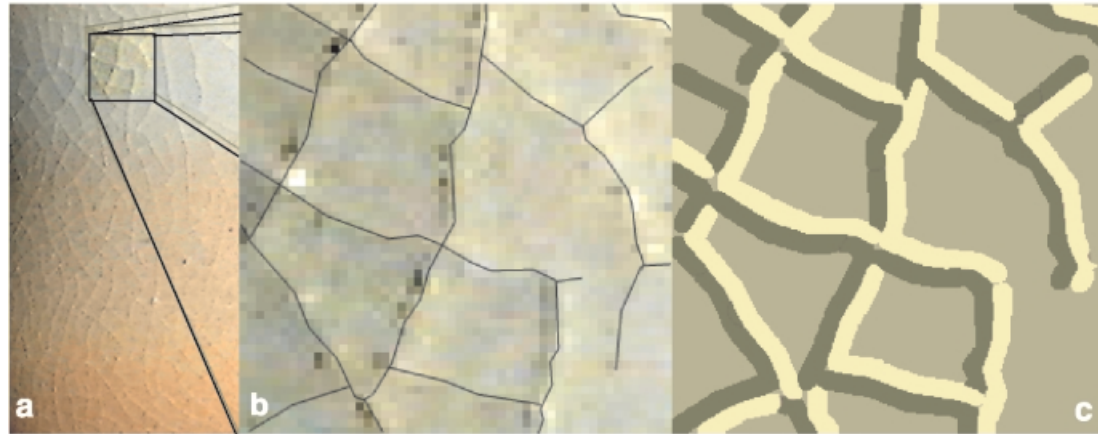
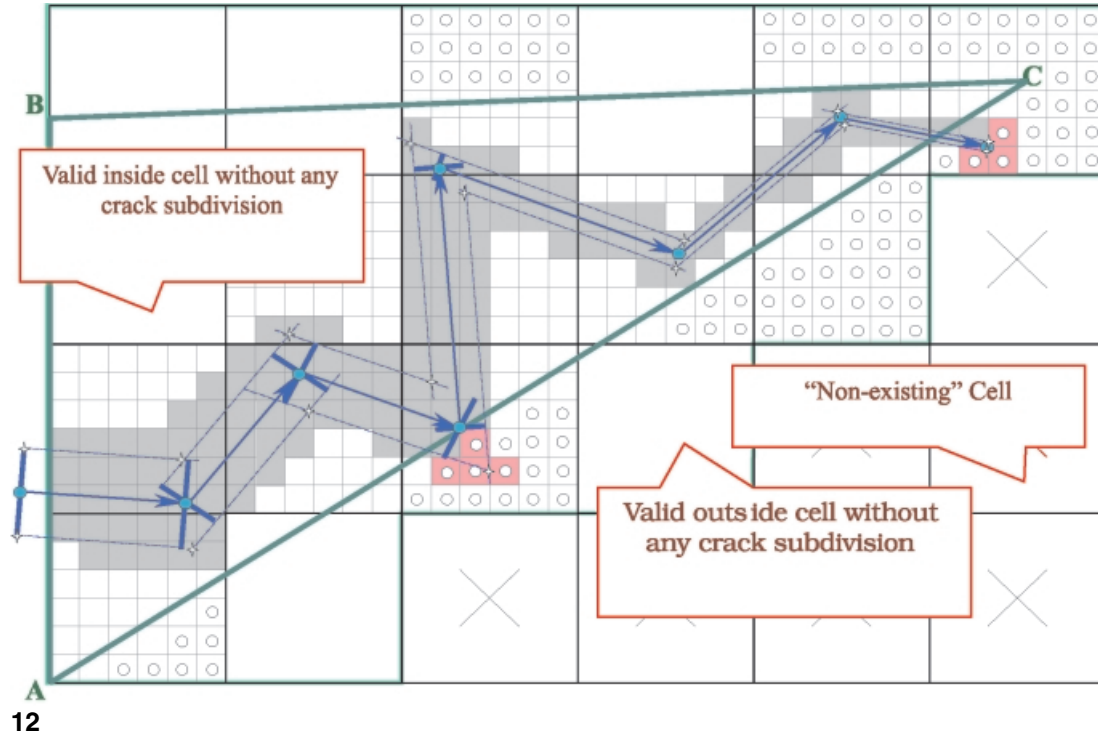
It is relatively complex to manage real 3D intersections of projected multiple cracks on 3D surface cellular layers. Therefore, for simulating this projection, we decided to use an approximation based on an anti-aliasing technique. It has the advantages of retaining good visual characteristics, while having a data structure which supports frequent changes. Our model subdivides the cracked cell, and we then use these micro-cells to apply our specific anti-aliasing model.

Table 2 (see Appendix B) summarizes the main steps for computing an anti-aliased linked-segment, projected on a triangle layer using line (segment) clippings. The sum of the lengths ( $\delta_i$ ) of the clipped segments gives a good approximation of the double integration needed to find the anti-aliasing.

Figure 12 shows an example of the resulting cell subdivision data structure. In this simple example, linked-segment thicknesses decrease constantly, cells are subdivided into  $6 \times 6$  micro-cells, and the triangle layer covers a cell array of only  $6 \times 4$ . The micro-cells in gray are those used for the anti-aliasing (not shown in this figure but see Plate 3). Note that the pink micro-cells do not belong to this triangle layer and thus must be ignored. In theory, the corresponding area should be computed in the neighboring triangle layer, but since these phenomena are extremely rare, we disregard them.

### 5.2 Inner-layer crack mirror effect special rendering

Sometimes the crack width is extremely thin, yet the path of the crack is obvious. This visual phenomenon

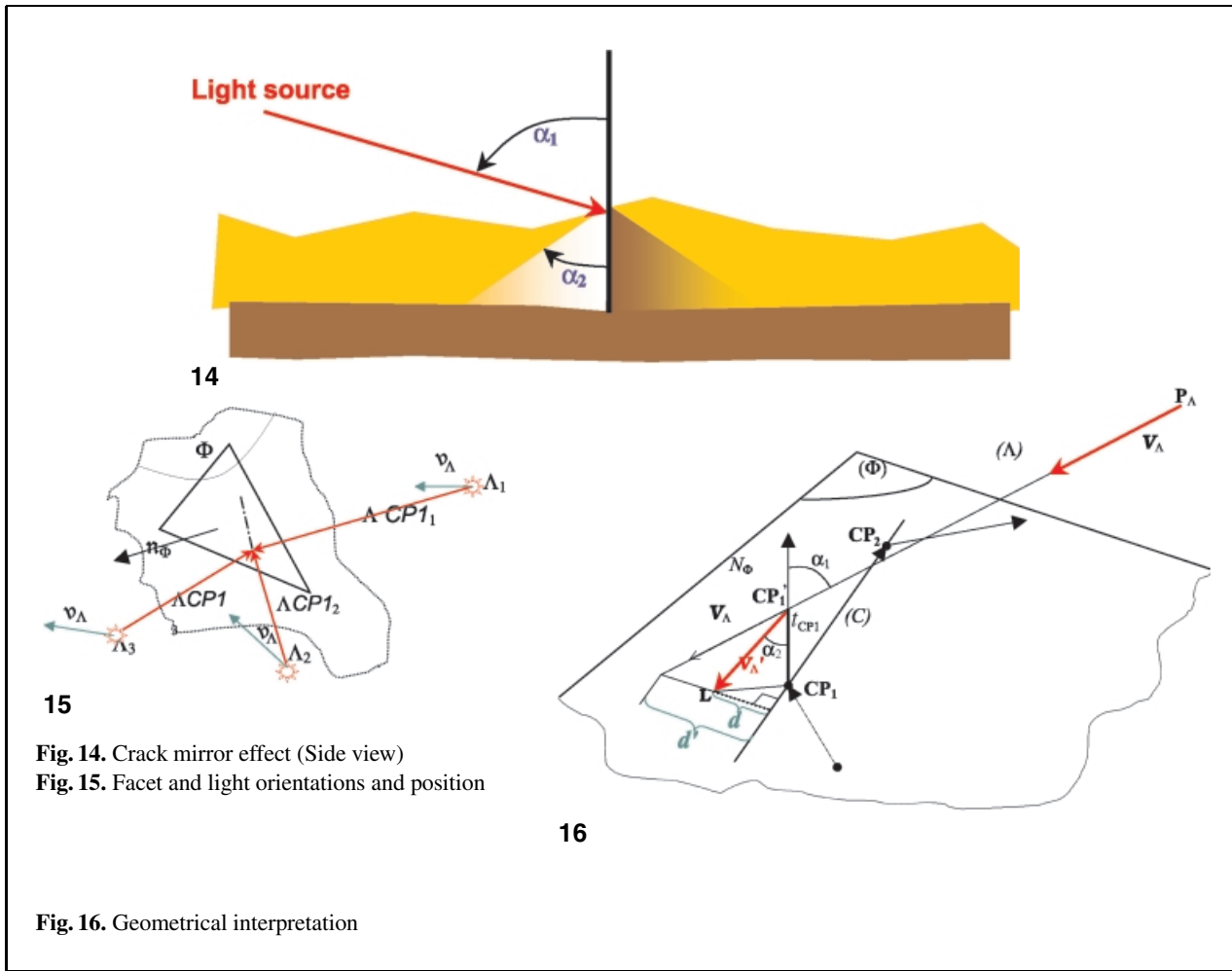


**Fig. 12.** Cell subdivision data structure

**Fig. 13.** Example of real ceramic mirror effect

is especially observed in the crack pattern of ceramic art. If we use our current model to render such a ceramic object, something seems to be missing: the crack thickness is very small, and the resulting patterns become almost invisible.

In this section we propose a possible extension of the crack pattern simulation using a 3D surface cellular automaton general model (Gobron and Chiba 1999) that specifically resolves the problem of the crack mirror special effect over multi-layer objects.



### 5.2.1 Inner-layer crack mirror effect

Figure 13 presents this effect, common in the field of ceramic art. Figure 13a shows an example of a crack pattern over the first glaze layer of a ceramic cup. The enlarged area (see Fig. 13b), shows that what appear to be actual cracks are in reality shadow and light reflections of cracks. Figure 13c presents a graphical interpretation of this phenomenon, the beige area and the dark areas being respectively the light reflections and the shadows. Figure 14 presents in detail a side view of the layer.

The refraction of light at the surface between two surfaces 1 and 2 is defined by Snell's law (1) with corresponding indices of refraction  $n_1$  and  $n_2$ :

$$n_1 \cdot \sin \alpha_{11} = n_2 \cdot \sin \alpha_2 \quad (1)$$

### 5.2.2 Resolving the crack mirror effect problem

To generate this mirror effect, we attempt to shade the surrounding of the cracks over the entire cellular surface using the multi-thickness linked-segment anti-aliasing method presented in Chiba et al. (1991). In completely shadowed regions, cracks appear not to exist. Therefore, the first task is to define the facet of the object on which this special effect occurs (see Fig. 15).

The conditions are then defined by:

$$\left. \begin{array}{l} \text{"Sun"} \\ \text{type lights} \end{array} \right\} \begin{array}{l} \Lambda_{CP1} \cdot n_{\Phi} < 0 \\ \{v_{\Lambda} \cdot n_{\Phi} < 0\} \end{array} \left. \begin{array}{l} \text{"Spot"} \\ \text{lights} \end{array} \right\}$$

Our main task consists of finding the symmetrical distance  $d$  of the light and shadow. We know that this distance is cast by the crack and projected onto the



surface of the second layer. Figure 16 shows the geometric interpretation:  
Equations (2) give a way to find the distance  $d$ .

$$\begin{aligned}
 C'_{P1} &= C_{P1} + N_{\Phi} \cdot t_{CP1} \\
 \alpha_2 &= \sin^{-1}(\sin \alpha_1 / \text{refraction Index}) \\
 &\Rightarrow V_{A'} = \text{rot}(V_A, (\alpha_1 - \alpha_2)) \text{ around } (N_{\Phi} \wedge V_A) \\
 (\Lambda') &= \begin{pmatrix} C'_{P1} \\ V_{A'} \end{pmatrix} \\
 L &= (\Lambda') \cap (\Phi) \\
 &\Rightarrow d = \text{projection} : L \mapsto (C)
 \end{aligned} \tag{2}$$

Note that the accurate computation of the refraction involves a very high computational cost (division, inverse sin, vector product, etc.). Thus we have studied a more efficient solution well adapted to the limitations of Computer Graphics.

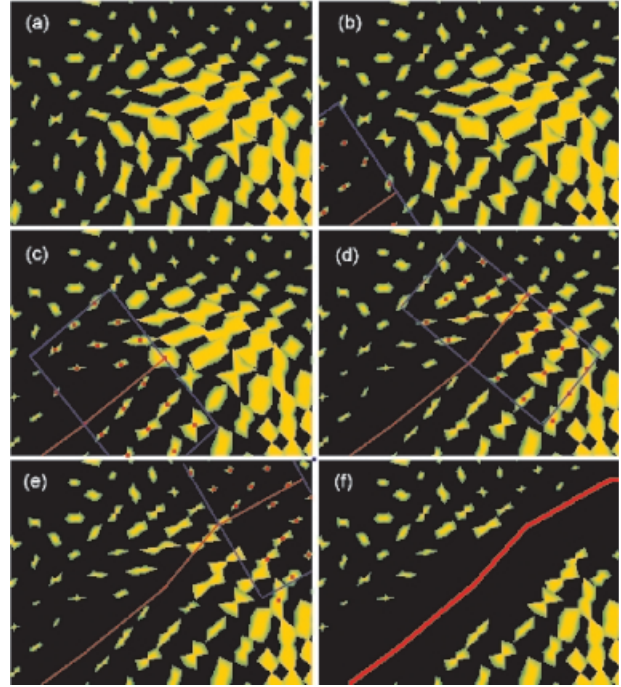
We first rewrite the problem as if no refraction phenomenon occurred – see  $d'$  in Fig. 16. Equations (3) show the solution to the simplified problem. Finally, to avoid computational errors on small angles and to simulate the refraction phenomenon, we deaden  $d$  into  $d'$  using for example, one of the two functions shown in (4).

$$\left. \begin{aligned}
 C'_{P1} &= C_{P1} + N_{\Phi} \cdot t_{CP1} \\
 \Lambda &= \begin{pmatrix} C'_{P1} \\ V_A \end{pmatrix} \\
 L &= (\Lambda) \cap (\Phi) \\
 d &= \text{projection} : L \mapsto (C)
 \end{aligned} \right\} \tag{3}$$

$$\left. \begin{aligned}
 d' &= T \cdot \frac{d}{1+d} \\
 d' &= T \cdot \frac{d}{\sqrt{1+d^2}}
 \end{aligned} \right\} \tag{4}$$

## 6 Results

Because the model is reasonably complex and a study of the rendering is not the purpose of this paper, we have chosen to present here a graphical summary of the main steps we have described in this paper, showing the behavior and possibilities of this crack model. This section is organized as follows: The first step is the pre-computation of the stress field. Plates 1 and 2 show the material elasticity and the resulting stress field over a relatively complex object. The second step – Plate 3 – presents the release of the



**Plate 3.** Progressive stress release parallel and inversely proportional to the distance to the crack path

stress contained in the material layers. Plate 4 deals with the final step: anti-aliasing and cell subdivision. Plate 5 demonstrates the main property of our model with the mutual interaction of two cracks. Finally, Plates 6 to 12 illustrate the possible resulting crack patterns using different types of materials (ceramic, mud, painting, glaze), some opaque and some semi-transparent, over the surface layer of simple or relatively complex objects.

All results – CA and graphical – except for the last plate were computed and rendered on an SGI Indigo2<sup>TM</sup> workstation, with 195 MHz CPU R10000, and 375 MB RAM; Plate 9 was generated using an SGI Origin with a 225 MHz CPU R10000 and about 1 GB RAM.

### 6.1 Stress field pre-computation

Plates 1 and 2 demonstrate our model's ability to simulate material elasticity over a tetrahedron and a 'complex' object (2017 input triangles, from SOFTIMAGE<sup>TM</sup> at <http://www.softimage.com>, subdivided into 412 029 cells with our 3D surface CA model). Image 2a presents the 2017 input triangles, and image 2b the corresponding stress field over the

object. Images 2c to 2f present four types of material showing a local stress field – from the upper part of the front right leg of the tiger (shaded image 2b). Image 2c being the original stress field (initially computed with the method described in Sect. 3.1), and from it we derive the elasticity of three other types of materials. In 2d and 2f, the cells of the material share 50% of their stresses with their neighbors, and in 2e and 2f the material repeats this sharing process 15 times (see Fig. 6). Therefore, 2d loses in intensity but not in frequency, which is quite difficult to see in the presented figure because, for graphical visibility purposes, stress spectrum intensities are attenuated by a logarithmic function for each rendering of a new object. In contrast 2e loses mainly in frequency, and we can especially notice the difference in stress field between the initial case 2c and the extreme case 2f: the former is very chaotic while the latter is much more harmonious.

## 6.2 Stress release

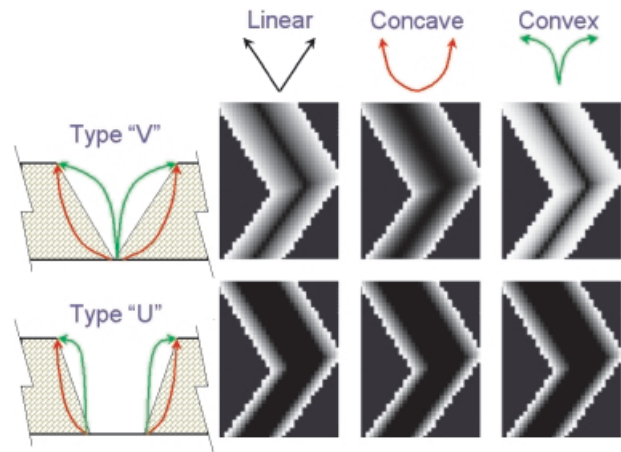
Plate 3 is the actual stress release of a region stress field due to the movement of a CM. Image 3a is the initial stress field. Images 3b to 3e are four steps (of nine in reality) of the CM animation. Finally, image 3f shows the elimination of the stress spectrum inside a 100% crack region.

Since the CM exists as a visible entity, we have superimposed on the scanned pictures some interesting data. The red linked-segment is the possible crack path in this direction; the red dots on the stresses indicate whether or not the stress spectrum has been affected by the release; blue rectangles represent the main areas where stress is released parallel to the crack's path and inversely proportional to the distance to the crack.

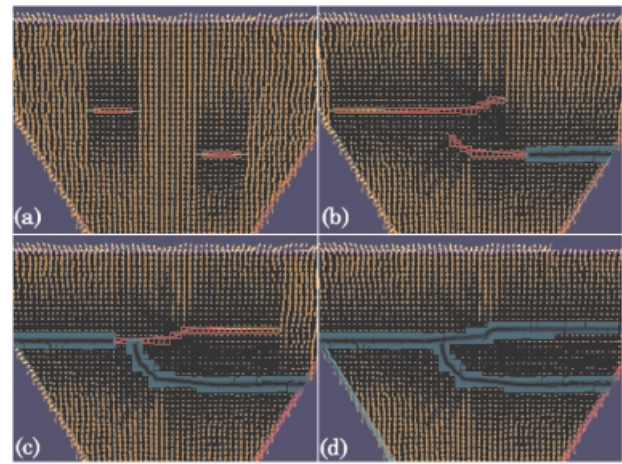
This method is essential for determining how the stress field is modified, as illustrated by the evolution from image 3a to 3e. Note that in 3a the set of stresses is more or less continuous in all directions making a harmonious field, while in 3b residual stresses near the crack are parallel to the path. Later (see Plate 5), because of this parallel arrangement, other CMs will change their directions in such a way that they make an angle of about  $90^\circ$ , depending on the CM velocity.

## 6.3 Anti-aliasing of linked-segment

Plate 4 presents the results of the anti-aliasing method summarized in Table 2 with a  $16 \times 16$



**Plate 4**



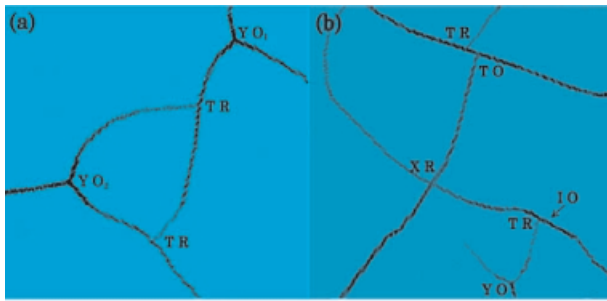
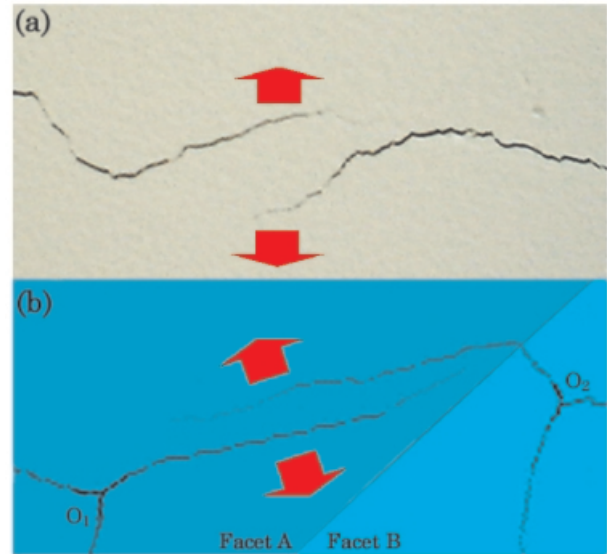
**Plate 5**

**Plate 4.** Several choices of anti-aliasing micro-cell subdivision  
**Plate 5.** Mutual interaction between two parallel type “I” cracks on a regular stress field

micro-cell grid. It shows six different types of anti-aliasing for simulating a linked-segment. The three result columns show the possible types of side-view curves: linear, concave, and convex. The first row presents direct result – type ‘V’. The second row presents a type ‘U’ central saturation – here 50% – making linked-segments looking darker (note the very light difference between pictures of this last row).

## 6.4 Mutual interaction between cracks

Plate 5 demonstrates our main goal, the mutual interaction between cracks on one facet of a tetrahedron with a regular stress field (pulling 90% up-down and

**Plate 6****Plate 7**

**Plate 6.** Two final crack propagations and renderings

**Plate 7a,b.** Typical realistic crack behavior: **a** crack picture; **b** our simulation

10% left-right). Image 5a shows two cracks of type “T” regularly releasing stresses on both sides. When the cracks move toward each other (see image 5b), their respective orientations (and hence velocities) are modified. In image 5c, the bottom crack is attracted increasingly rapidly to the crack above, generating approximately a right angle. At the same time the above crack, which was first driven away, returns to its main course (driven toward the main intensity of the stress field). The last picture shows the final crack pattern. (Note that in Plate 5, the cyan region indicates the automatic anti-aliasing micro-cell.)

## 6.5 Final results

Plate 6 presents two selected regions of crack propagation where minor (parasite) cracks have been re-

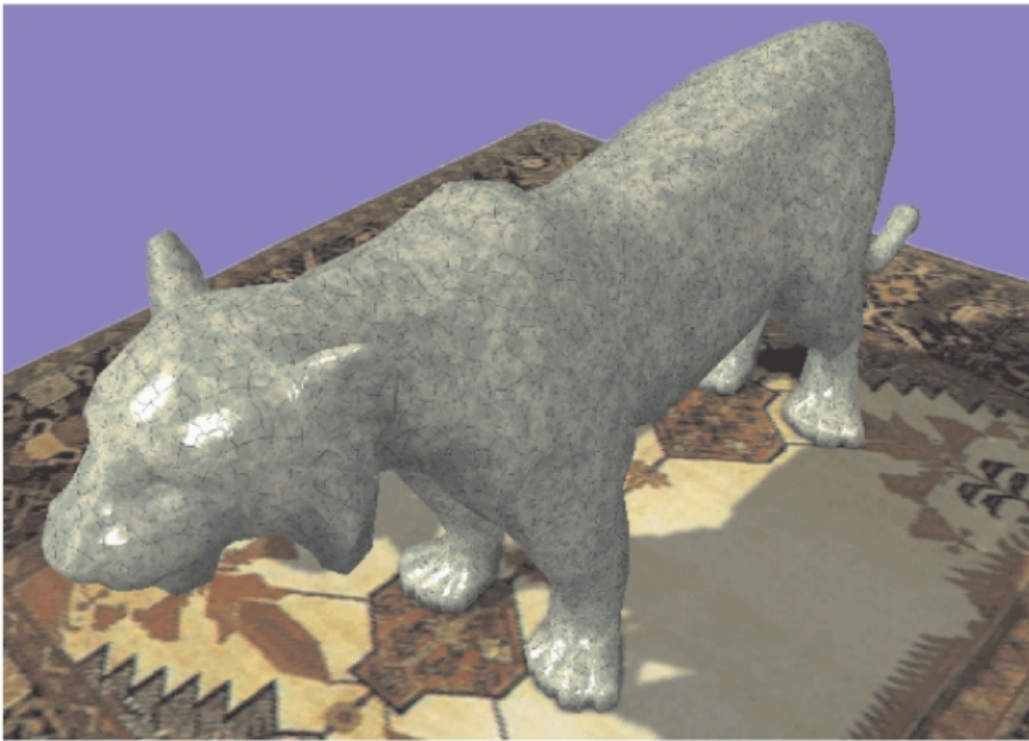
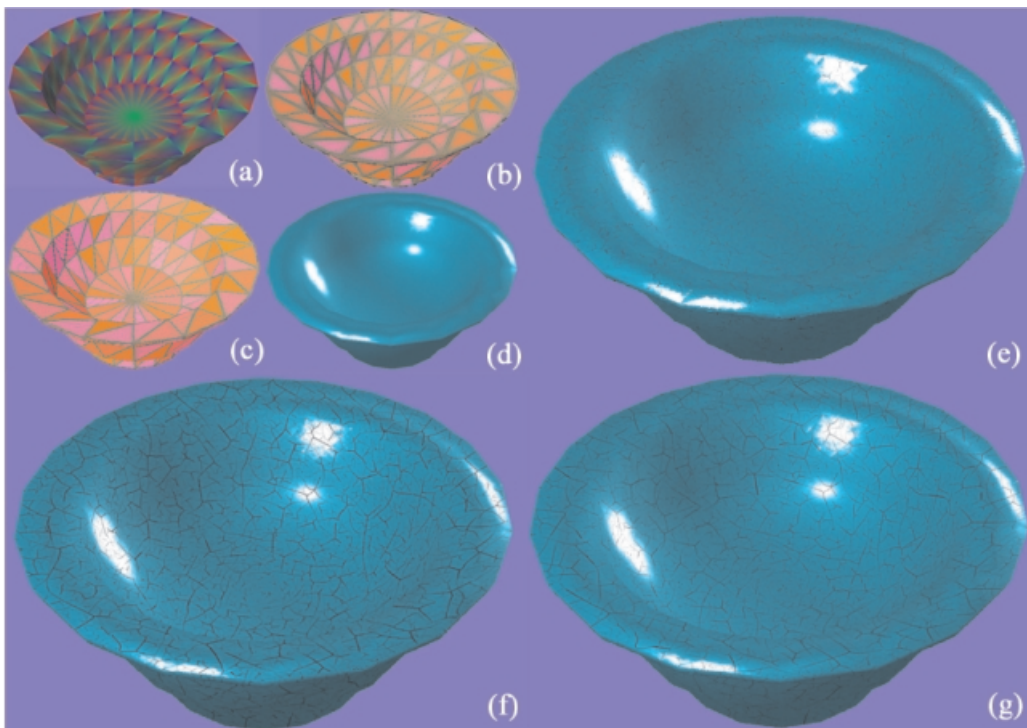
moved. In both images, crack origins indicated by an “O”, and the resulting crack intersections by an “R”. On the one hand, image 6a shows a simple case where just two “Y” type cracks are propagating their respective three CMs, resulting in two simple “T” crack patterns. Note that the mutual attraction between the CMs – due to the progressive modification of the stress field – generates realistic “T” crack patterns. On the other hand, image 6b involves three different crack origins (“T”, “T”, and “Y” types). The resulting pattern is much more complex: One of the “T” crack CMs crossed at the same time the “I” type crack making an “X” crack pattern; this phenomena is very rare. Later a “Y” type crack was produced. Its right branch reached the path of the “I” crack, but since its left branch was approaching the “X” crack pattern, it rapidly lost all its inertia and stopped before reaching another crack path.

Plate 7 presents a comparison between a real picture in 7a, and one of our simulation in 7b, where both images show a crack discontinuity. This special crack behavior occurs relatively frequently in Nature and is due to the simultaneous release of a stress field that has a single main direction (here vertical to the figure: see red arrows). The same phenomenon occurred during our simulation. Two “Y” cracks ( $O_1$  and  $O_2$ ) were produced at about the same time on facets A and B. The CMs approaching each other released the same stress field, and then progressively stopped due to a lack of inertia and other releasable directional stress.

Plate 8 presents the rendering of a tiger made with two layers. The crack pattern covered by a layer of glaze is simulated over the second layer of gray concrete. What is particularly interesting in this plate is that cracks are omnipresent: we do not really notice them but if they were not shown, the object would appear completely different.

The main data used for this simulation were: the 2017-triangle tiger was decomposed this time into over a million cells, each cracked cell was subdivided into  $3 \times 3$  micro-cells; the elasticity of the material was 25% on a maximum range of 15 cells, and the CMs were influenced by 75% of inertia. Note also that the stress field was irregular, that the total stress released was about 100%, and that the total pre-computational time (including pre-computation, crack pattern computation, and floor shadow casting) was nearly 2 h, but that each image renderings computation took less than 3 min (including floor texture).



**Plate 8****Plate 9**

**Plate 8.** Crack propagation over a 'complex' object first layer; note that stress was released at only 35%

**Plate 9.** Identical stress field and material, different crack parameters

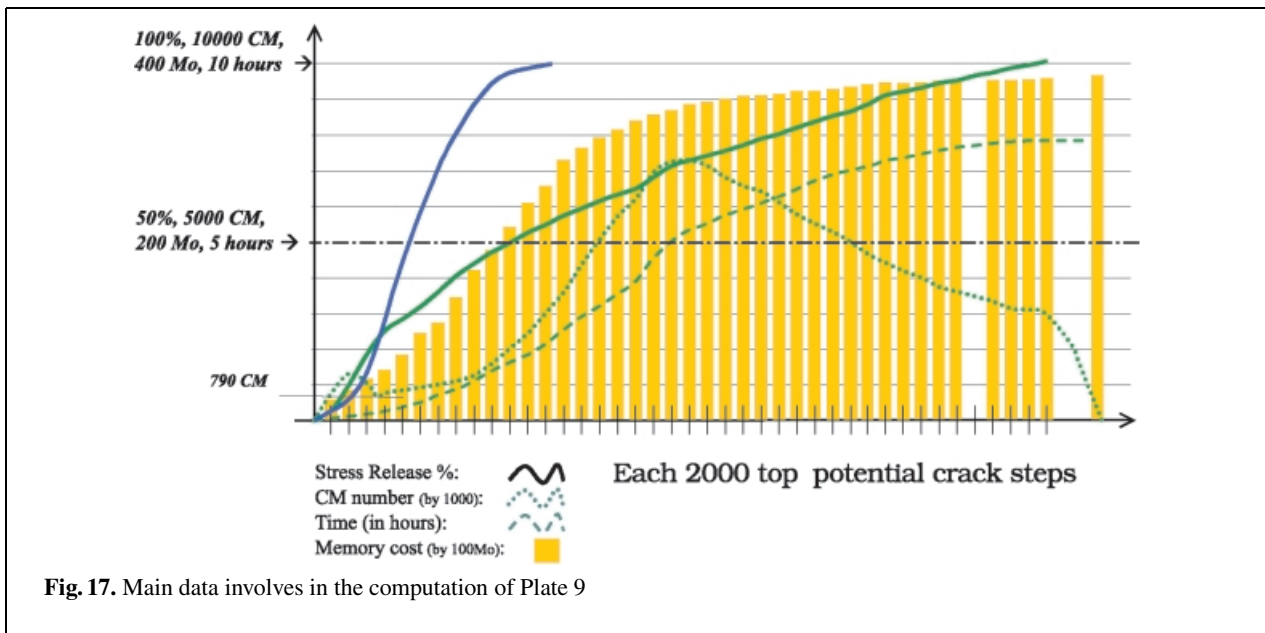


Plate 9 presents another ceramic object with different crack and material parameters. Image 9a depicts the object's 480 input facets with their respective RGB orientations (see Gobron and Chiba 1999), and images 9b and 9c show the corresponding 98 224 and 477 006 cells (in beige). The two stress fields over the objects were artificially set so that the one in 9c was four times larger than that in 9b. Only the corresponding stress spectrums inside the triangle facet are also shown in mauve to pink shadings (to differentiate facets).

After material stress relaxation the initial number of unstable cells was 28 634 for 9a and 198 440 for 9b. The objects before crack propagation were pre-rendered and in both cases resulted in a similar image, 9d. Finally, we set the micro-cell sizes in both simulations so that the pixel definition of the final images would be the same.

The resulting crack patterns are very different. Image 9e presents the final crack pattern of 9b and shows a large number of cracks that are very similar to each other but almost invisible. In contrast, image 9f shows long and thick cracks (the majority being of "Y" type) as well as a very large number short and thin "T" and "I" type cracks. In image 9g, two parameters were different to those of 9f: we made the  $S_{ii}$  constant 10 times smaller and we decreased  $m_{cw}$ . The resulting pattern is quite similar to 9f but shows longer and thinner major "Y" type

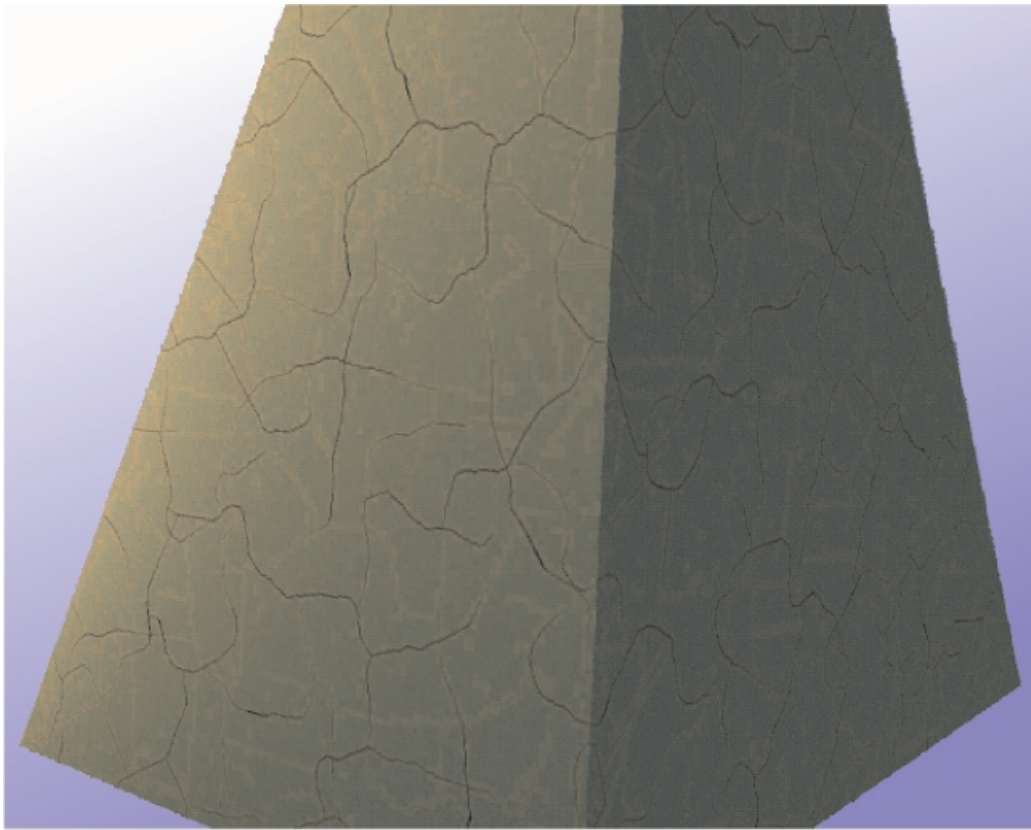
cracks. Figure 13 shows the main data behavior that occurs during the computation of the final image 9g. We note in particular that its computation required about 400 Mb of memory, that the maximum number of simultaneous CMs was about 7300 (with a local minima of 790, see Fig. 17), that crack propagation was fast at the beginning and at the end, and that the total computational time was approximately 8 h. To make possible a comparison with a smaller scene, the blue curve represents the stress release percentage that occurs in 9e.

The last image of this section – Plate 10 – presents the visual simulation of a mud crack pattern over the 3D Surface of a tetrahedron. This approximately 330 000 cell simulation was obtained by setting the micro-cell grid to  $11 \times 11$ ; 85% of the cells were initially unstable; the stress relaxation was only of 5%; the elasticity recursion was 25 cells; and finally the CMs had a 65% inertia influence. The object reached 99% stability after 20 min of computation. An interesting factor in this simulation is that we set the crack-seeds to be only of type "I" cracks, and yet our model naturally simulates many "Y" intersections.

## 6.6 Crack mirror effect results

Plates 11a to c present a the view of a single facet with three different light positions. The materials used for this simulation were a thick layer of green





**Plate 10.** Visual simulation of mud crack pattern over a tetrahedron using only “type I” cracks

ceramic (underneath) and a thin layer of transparent glaze where cracks have been propagated during pre-computation (Chiba et al. 1991). The crack pattern is clearly visible through the cracks’ shadows and light effects. We can also see the opposite orientation of the mirror between Plate 11a and 11c, and the appearance and disappearance of some cracks depending on the light orientation. In this example, computation times were a few minutes per images. Plate 13 shows the rendering of the ceramic crack mirror effect over a more complex object. There were 2017 input triangles for this tiger, the total number of cells was about one million, the number of cracks about 100 000, the shadow/light subdivision about 4.5 million, and the total computation time nearly 3 h.

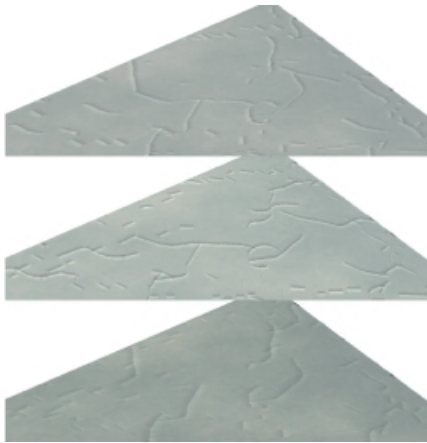
## 7 Discussion

We have presented a convenient model for generating crack patterns on any type of 3D object using

a method based on multi-layer CA. We explained the need for a semi-physical approach and proposed an intuitive but efficient stress spectrum model to simulate and easily compute the stress field over any layers of the 3D surface Cellular Automaton. Then we detailed how crack patterns were generated using the stress released by crack modules. To easily improve the appearance of the resulting images, we used an original linked-segment anti-aliasing method, which was also used for the release of the stress field. We verified our model’s performance by presenting the resulting crack patterns of the four main steps used to generate the 3D surface crack propagation based on CA. Finally, we presented a series of rendered images showing several types of cracks: concrete, ceramics, mud, and glaze with its typical special mirror effect.

This work could be further improved by considering real interaction between layer models and better 3D surface CA.

Many aspects of the layer crack model could be improved. For example, we could generate more

**Plate 11****Plate 11.** Crack mirror effect: changing the light positions**Plate 12****Plate 12.** Example of crack mirror effect on a complex object

accurate cracks in 3D using local-Hypertexture (especially of mud), improved the stress release model, render higher-quality crack patterns [see super-sampling (Westin et al. 1992)], include the interactions between layers, produce an even better solution to the velocity relationship between cracks, or develop mixed types of material for a layer, and therefore developing the crack subdivision for high tension absorption simulating of composite (non-pure) materials.

Improving 3D surface CA will probably lead to the study of linking this 3D surface CA model to non-grid based models, such as a water flow system [as

suggested in Dorsey et al. (1996)]. This association of models would generate realistic corrosion and the corresponding patina paintings after the peeling of a covering layer previously cracked by the model summarized in this paper.

*Acknowledgements.* We thank Norihiro Nasukawa for his system assistance and Batja Sosorbaram Batzul for her mathematical services. Special thanks are also due to Corinne Kapel, William A. Brannen, and Tonya Tomiyama for their daily support and their helpful comments on drafts of this paper.

This work was supported partly by "A Support System for Region-specific R&D Activities" of the Telecommunications Advancement Organization of Japan.

## Appendices

### A. Main parameters of our crack simulation model

Please see Table 1.

### B. Anti-aliasing multi-thickness linked-segment

Please see Table 2 on the next page.

### C. Pseudo-code of the crack algorithm

- 1 Generate the 3D Surface Multi-Layer Cellular Automaton object pre-analysis, cellular pre-computation, edge communication, layer-surface generation (see Gobron and Chiba);

**Table 1.** Material parameters

Categories	Symbols	Short definition
Material	$e_L$	Elasticity loop range
	$e_P$	Elasticity percentage
	$m_R$	Resistance maximum stress
	$S_{ii}$	Stress intensity interval
	$m_{cw}$	Maximum crack width
Crack types & Crack Modules	$CS_{at}$	Crack segment angle tolerance
	$CS_a$	Crack segment min angle
	$I_{ip}$	Inertia influence percentage

- 2 Pre-compute stress field over all layer depending on material properties and layers thickness surfaces;
- 3 Pre-compute the material self-relaxation;
- 4 Determine scaling stress field factor for visualization of the stress spectrum (optional);
- 5 Crack generation:
  - 5.1 Fill and sort first unstable cell list, i.e., where total cell material resistance is less than at least one of the stress spectrum directional intensities: Pot-Crack list;
  - 5.2 Determine  $M$  and  $m$  maximum intensities respectively for the first and last stress-spectrums of the Pot-Crack list;
  - 5.3 Compute crack propagation until both Pot-Crack and CM lists are empty;
    - 5.3.1 Set  $t = M$ ;
    - 5.3.2 If (Pot-Crack list not empty) **do**:  
(Check if a new series of crack set must be called);  
(Subtract  $S_{ii}$  from  $t$ );  
**While** CM list is empty and if Pot-Crack list is not empty.
    - 5.3.3 Re-order the CM list;
    - 5.3.4 Move all CMs, with collision test, CM facet-layer 3DSCA data transfer, stress release, and if CM elimination or change of facet, compute and set anti-aliasing of CM path through layers;
    - 5.3.5 If (Pot-Crack list empty) **do**:  
(Try to refill Pot-Crack list), then (if succeed) **do**:  
Reset  $M$ ,  $m$ , and  $t$ ;
  - 5.4 Release dynamic memory allocations.

**Table 2.** Main steps for anti-aliasing multi-thickness linked-segments on a triangle layer

Procedures/Methods	Graphical interpretations
1 • LOOP <sub>[i]</sub> for all segments $S_i$ Call step 2	
2 • FIND $i$ th thickness segment vectors $V_{start}$ and $V_{end}$ • CALL step 3	
3 • FIND integral-segments coordinates $\Phi_{start}$ and $\Phi_{end}$ • LOOP <sub>[i]</sub> for all of them: CALL step 4	
4 • FIND anti-aliasing segment • LOOP <sub>[k]</sub> sub-segment $\phi_k$ from $\Phi_{start}$ to $\Phi_{end}$ . IF the $\phi_k$ inside non-subdivided cell $C$ : CALL step 5	
• FIND micro cell $\mu_k$ at $\phi_k$ position • FIND length $\delta_l$ by CLIPPING $\phi_k$ on $\mu_k$	
5 • SUBDIVIDE $C$ into micro-cells. • IF the triangle contour crosses the MACRO- cell: CALL step 6	
6 • UNVALID outside triangle contour micro-cell of $C$	

## References

1. Barzel R (1997) Faking dynamics of ropes and springs. *IEEE Comput Graph Appl* 17(3):31–39
2. Broek D (1991) *Elementary engineering fracture mechanics*, fourth revised edn. Kluwer Academic Publishers
3. Chaudhuri PP, Chowdhury RD, Nandi S, Chattopadhyay S (1997) Additive cellular automata theory and applications V1. *IEEE Comput Soc Press*
4. Chiba N, Wada S, Kaino K, Muraoka K (1991) A behavioral model of cracks and its applications to CG. *Systems Comput Japan* 22(11)
5. Codd EF (1968) *Cellular automata*. Academic Press, New York London
6. Dana KJ, van Ginneken B, Nayar SK, Koenderink JJ (1999) Reflectance and texture of real-world surface. *ACM Trans Graph* 18:1–34
7. Davidge RW (1979) *Mechanical behaviour of ceramics*. Cambridge Sol State Sci Ser
8. Dorsey J, Hanrahan P (1996) Modeling and rendering of metallic patinas. *Proceedings of the SIGGRAPH conference*, pp 387–396
9. Dorsey J, Pedersen HK, Hanrahan (1996) P flow and changes in appearance. *Proceedings of the SIGGRAPH conference*, pp 411–420
10. Fleischer KW, Laidlaw DH, Currin BL, Barr AH (1995) Cellular texture generation. *Proceedings of the SIGGRAPH Conference*, pp 239–248
11. Fowler R, Meinhardt H, Prusinkiewicz P (1992) Modeling seashells. *Proceedings of the SIGGRAPH Conference* 26:379–387
12. Gobron S, Chiba N (1997) Visual simulation of corrosion. *Workshop of Tohoku'97*, Morioka, Japan, ref 97-3-9
13. Gobron S, Chiba N (1999) 3D surface cellular automata and its applications. *J Vis Comput Anim* 10:143–158
14. Hirota K, Tanoue Y, Kaneko T (1998) Generation of crack patterns with a physical model. *Visual Comput* 14:126–137
15. Hirota K, Tanoue Y, Kaneko T (2000) Simulation of three-dimensional cracks. *Visual Comput*
16. Hsu SC, Wong TT (1995) Simulating dust accumulation. *IEEE Comput Graph Appl* 18–22
17. Ito H, Miyata Y (1998) Experimental study on mud crack patterns (in Japanese). *J Geol Soc Japan* 104:90–98
18. Mazarak O, Martins C, Amanatides J (1999) Animating exploding object. *Proceedings of the Graphics Interface Conference*, pp 211–217
19. Neff M, Fiume E (1999) A visual model for blast waves and fracture. *Proceedings of the Graphics Interface Conference*, pp 193–202
20. Neider J, Davis T, Woo M (1997) *OpenGL programming guide*, 2nd edn. SGI, Addison-Wesley Publishing
21. Norton A, Turk G, Bacon B, Gerth J, Sweeney P (1991) Animation of fracture by physical modeling. *Visual Comput* 7:210–219
22. O'Brien JF, Hodgins JK (1999) Graphical modeling and animation of brittle fracture. *Proceedings of the SIGGRAPH Conference*, pp 137–146
23. Perlin K (1989) Hypertexture. *Proceedings of the SIGGRAPH Conference*, pp 253–262
24. Preston K, Duff MJ, Leviadi S, Norgren PE, Toriwaki JI (1979) Basics of cellular logic with applications in medical image processing. *Proceedings of IEEE*. IEEE Press, Piscataway, NJ, p 67
25. Rosenfeld A (1979) *Picture languages*. Academic Press, New York
26. Skjeltorp AT, Meakin P (1988) Fracture in microsphere monolayers studied by experiment and computer simulation. *Nature* 335:29
27. Stanley EH, Ostrowsky N (1986) *On growth and form*. Martinus Nijhoff Publishers
28. Sternberg SR (1980) *Language and architecture of parallel image processing*. North Holland, Amsterdam
29. Takai Y, Ecchu K, Takai NK (1995) A cellular automaton model of particle motions and its applications. *Visual Comput* 11:240–252
30. Terzopoulos D, Fleischer K (1988) Modeling inelastic deformation: viscoelasticity, plasticity, fracture. *Proceedings of the SIGGRAPH Conference* 22:269–278
31. Thalmann D (1986) A 'lifegame' approach to surface modeling and rendering. *Visual Comput* 384–390
32. Turk G (1991) Generating texture for arbitrary surfaces using reaction-diffusion. *Proceedings of the SIGGRAPH Conference*, pp 289–298
33. Westin SH, Arvo JR, Torrance KE (1992) Predicting reflectance functions from complex surfaces. *Proceedings of the SIGGRAPH Conference*, 289–298
34. Wong TT, Ng WY, Heng PA (1997) A geometry dependent texture generation framework for simulating surface imperfections. *Proceedings of the 8th EuroGraphics Workshop on Rendering*, pp 139–150

Photographs of the authors and their biographies are given on the next page.



STÉPHANE GOBRON is a doctoral student in the Department of Computer Science at Iwate University, Morioka, Japan. He received a BS in computer graphics and software engineering from the Florida Institute of Technology, FL, U.S.A. and DEA (Master) in computer graphics and CS languages theories from the University of Nancy I, Loria, France, in 1994 and 1995 respectively. In 1996, he worked as a computer engineer responsible for CG concep-

tion and design. His research interests in computer graphics include natural phenomena (especially CG aging, weathering, and automatic texturing) and cellular modeling (using multi-layers and 3D surface techniques). He is a member of ACM and IEEE.



NORISHIGE CHIBA is currently a Professor in the Department of Computer Science at Iwate University. His research interests include computer graphics, algorithm theory, and science on form. He received a BE in electrical engineering from Iwate University and an ME and DE in information engineering from Tohoku University in 1975, 1981, and 1984, respectively. He worked at Nippon Business Consultant Co., Ltd from 1975 to 1978. He was a Research Associate in the Department of Communication

Engineering at Tohoku University from 1984 to 1986, an Associate Professor of Computer Science at Sendai National College of Technology from 1986 to 1987, and an Associate Professor of Computer Science at Iwate University from 1987 to 1991. He is a member of IEICE Japan, IPS of Japan, IEEE and ACM.