A Corrosion and Deformation Simulation Method of 3D Iron Objects based on Voxel Automaton

Yuta Suzuki †
Yuta Yamabe †
† Tokyo Denki University Graduate School,
Tokyo, Japan
y-suzuki@vcl.jp †
y-yamabe@vcl.jp †

Tomoaki Moriya ‡
‡ Tokyo Denki University,
Tokyo, Japan
moriya@vcl.jp ‡

Tokiichiro Takahashi ‡ *
*ASTORODESIGN, Inc.
Tokyo, Japan
toki@vcl.jp ‡ *

Abstract—We propose a visual appearance generation method of corrosion and deformation of 3D iron objects based on a voxel automaton, an extended 2D cellular automaton to 3D, that generates rusted 3D models. Our method enables to generate progress of rust, which often causes surface roughness on iron object, and eventually opens a hole in the 3D iron objects due to corrosion.

Keywords—corrosion; rust; voxel; cellular automaton; aging; computer graphics

I. Introduction

In recent years, improvement in expressive power of 3D computer graphics is remarkable. It is possible to express various objects made of various materials with 3DCG. Many objects existing in real space may be changed in color and deformed their shapes significantly due to long-term exposure to sunlight, rain, wind, sand, atmosphere, and so on. These phenomena are called '*aging*' or '*weathering*'.

Corrosion is one of the most familiar aging phenomena of metals. Corrosion is a reaction that causes by contact of water with base metals such as iron. As corrosion progresses, iron becomes fragile. Eventually some parts of iron are peeled off. As a result, iron surfaces become irregularly bumpy. Rust proceeds further on the irregularly bumpy surface, and may proceed until a hole is made in the iron plate. Corrosion due to rust often pierces the iron plate.

Conventional methods [1]-[5] have simulated discoloration and irregularities on the iron surfaces due to corrosion. However, it is difficult to simulate further progress of rust corrosion which penetrates an iron plate by rust peeling off.

In this paper, we propose a rust progression simulation method including penetration of an iron plate based on voxel automaton.

II. Previous Work

A. L-system Based Approach

Yao-Xun Chang et al. [1] and Kamata et al. [2] tried to simulate rust progression using the L-system. Parameters regarding rust progression are given to each vertex of 3D polygonal models. These parameters are updated by L-system. As a result, surfaces of the 3D shape models become irregular and bumpy. However, this approach has a weak point. It is impossible to express a phenomenon that a hole penetrating from the front surface to the back surface of the 3D shape model opens.

B. 2D Cellular Automaton Based Approach

Doi et al. [3] proposed an automatic generation method for rust texture images using a two-dimensional cellular automaton. Each cell holds five values: vertex coordinates, surface attributes, growth level, phase level, and rust occurrence probability. Growth level expresses the degree of rust progression of each cell by positive integer. Phase level is a value determined by the degree of rust progress. Rust occurrence probability can adjust the occurrence probability of rust arbitrarily given in advance. The growth level of each cell propagates to neighboring cells. As a result, the phase levels of neighboring cells rise, and eventually, the rust is spreading.

C. Main/Sub-Rust Based Approach

Tanabe et al. [4] and Kanazawa et al. [5] have generated rust texture images on an iron plate by using a two-dimensional cellular automaton. Both of two methods introduced two different kinds of rust, such as main-rust and sub-rust. The main-rust is produced by iron corrosion, and the sub-rust is rust which is detached from the main-rust due to rain and wind. By introducing the main-rust and sub-rust, they have succeeded in generating very realistic rust as texture images. In addition, their method can change rust propagation speed according to the tilt angle of the iron plate between the ground (horizontal plane).

In these previous studies [1]-[5], it is possible to generate realistic texture images of rust, and to express irregularities of rusted models. However, these methods cannot represent that corrosion penetrates a hole on iron plates.

III. Proposed Method

In order to represent that corrosion penetrates a hole on iron plates, we propose and extend the rust progression methods [4] [5] based on a two-dimensional cellular automaton to a three-dimensional voxel automaton.

A. Data Structure of Voxel

Data structure of each voxel v_i is as follows:

1. Positions $\vec{p_i} = (x_i, \ y_i, \ z_i)$
2. Tilt angle θ_i
3. Degree of rust progression d_i
4. RGB color values $\vec{c_i} = (r_i, \ g_i, \ b_i)$

978-1-5386-2615-3/18/$31.00 ©2018 IEEE

As each parameter changes, rust progresses. RGB color values $\vec{c_i}$ of each voxel changes in accordance with the value of the rust progression degree d_i as shown in Table 1.

B. Occurrence of Rust

The rust progression degrees d_i of all voxels are set to 0 at the start of the simulation. Voxels on surfaces of a 3D shape model imitating an iron square pipe generate rust with a certain probability when the iron square pipe contacted with water due to oxygen dissolved in water. In this study, we consider only iron rust due to contact between water and dissolved oxygen. The rust progression degree d_i of each voxel v_i increases by 1 where rust occurred. The incidence rate of rust is 0.001% that as with the previous methods of Tanabe et al. [4] and Kanazawa et al. [5]. Initial rust is generated by this processing.

TABLE I. CORRESPONDENCE BETWEEN d_i AND $\vec{c_i}$ OF VOXEL v_i

Degree of rust progression d_i	RGB Color $\vec{c_i}$
0	
20	
35	
45	
55	
65	
75	
85	
≥ δ	penetrate

C. Simulation of Rust Progression and Diffusion based on Voxel Automaton

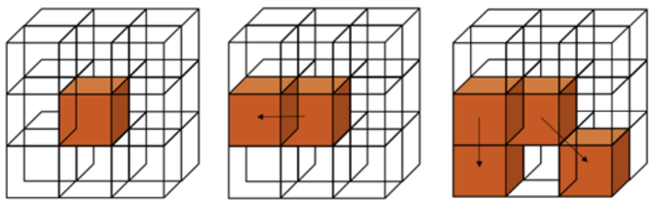
The steps of 'rust propagation and diffusion process' based on our voxel automaton method are as follows. It is defined as 1 loop that once the following steps from 1 to 3 are executed:

A voxel v_i whose rust progression degree $d_i \geq 1$ is called a rust voxel. Rust propagates from the rust voxel v_i to the connected voxels thereof to become rust voxels. Then, rust propagates from each voxel that has become rust voxel. As a result, the rust voxels diffuse.

1. Considering the tilt angle θ_i of the rust voxel v_i, a voxel v_j is selected as the rust propagation direction from neighbor voxels $\{u_i\}$ of the rust voxel v_i.
2. Then, the rust propagation degree d_i of the rust voxel v_i is incremented. Its colors $\vec{c_j}$ is updated to the color corresponding to the rust progression degree as already shown in Table 1.
3. If the rust progression degree $d_i \geq \delta$, the voxel v_i is deemed to be detached from the iron object, i.e., the voxel v_i is deleted with certain probabilities. Here, δ is a threshold value (δ = 90).

This loop is iterated, then the rust propagation and diffusion processes are realized.

Fig. 1(a) give an example of a rust voxel on the surface of the 3D shape model and 17 voxels in vicinity of the rust voxel. Fig. 1 (b) is a pattern diagram showing how rust propagates from a rust voxel to neighbor voxels at Step 1, then one of the neighboring voxels becomes a new rust voxel. The first rust voxel as shown in Fig.1 (a) diffuses by the rust propagation to the neighboring voxel (see Fig.1 (b)), then propagate again (see Fig.1 (c)).



(a) First rust voxel (b) Rust propagation once (c) Rust propagation twice

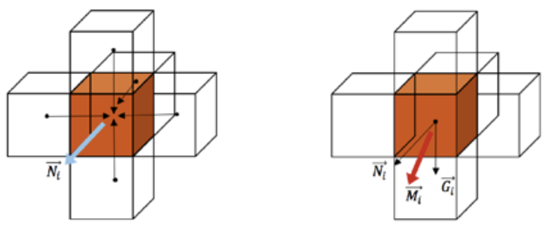
Fig. 1. Propagation of rust of voxels

D. Determination of propagation direction

In this section, the mechanism of determination of propagation direction described in the previous section. For the sake of simplicity, we describe it in the case of the number of neighboring voxels is 5 as an example. Actually, the number of neighboring voxels changes from 1 to 26 depending on the shape of the 3D object to be produced. The determination steps are as follows:

1. Calculate all the vectors from one of the neighboring voxels $\{u_i\}$ to its central rust voxel v_i. Furthermore, we calculate the sum vector of them. Let the sum vector be the normal vector $\vec{N_i}$ of rust voxel v_i (Fig. 2 (a)), where the normal vector $\vec{N_i}$ is a unit vector.
2. We define the gravity vector $\vec{G_i}$ that vertically lowered from a rust voxel v_i in the direction of the gravity. The gravity vector $\vec{G_i}$ is a unit vector. We calculate the sum of the normal vector $\vec{N_i}$ of the rust voxel v_i and the gravity vector $\vec{G_i}$. The summed vector $\vec{M_i}$ is converted to a unit vector (Fig. 2 (b)).
3. One of the neighboring voxels $\{v_k\}$ which includes the unit vector $\vec{M_i}$ is selected as a voxel that rust most easily propagated from the rust voxel v_i.

In this method, the gravity vector $\vec{G_i}$ is always pointing the direction of the Y axis in the world coordinate system. Accordingly, the rust propagation taking inclination into account can be done even if the voxels are tilted.



(a) Normal vector of a rust voxel (b) Calculation of the most easily rust propagation direction

Fig. 2. Rust propagation direction

IV. Experiment Results and Evaluation

We have executed some experiments to evaluate the proposed method whether it can generate realistic rust progression on the iron plate as well as corrosion which penetrates a hole on the iron plates.

A. Rust progression results

We prepared three kinds of the voxel data as indicated in Table 2, and their profiles are shown in Fig. 3, Fig. 4, and Fig. 5, respectively. Fig.3(b), Fig.4(b), and Fig.5(b) are after iterating 200 times of the rust progression steps from Step 1 to Step 3, described at Section III-C, respectively.

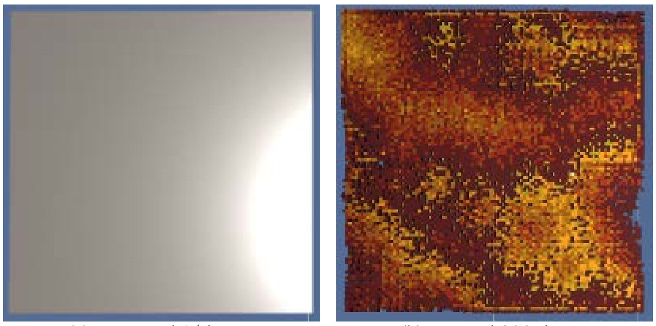
Fig.3 is an example of a voxel representation which models a horizontally placed iron plate. We can observe that rust propagates along with the rust progression steps.

TABLE II. VOXEL DATA

Voxel data		Fig. 3	Fig. 4	Fig. 5
Imitated objects		An iron plate	A square pipe	An iron ball
Number of voxels	X axis	100	4	50 (Max)
	Y axis	3	100	50 (Max)
	Z axis	100	4	50 (Max)
	Total	10,000	1,600	40,000
Tilt angle		0°	60°	0°
Projection direction		Y axis	Diagonally above 45°	Diagonally above 45°

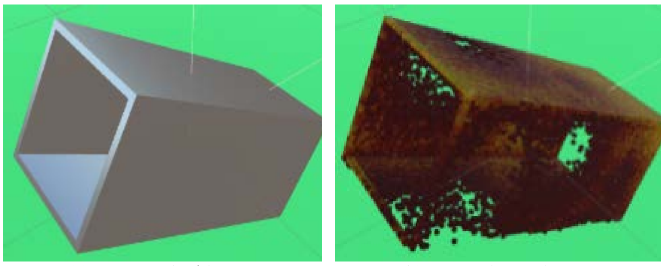
When the whole voxels are tilted (see Fig. 4), rust propagates along the tilt. We also can see the holes were opened after 200 times iteration.

In the voxel representation of an iron ball (Fig. 5), it can be seen that the voxels on the outer surface are peeled off and the inner voxels are exposed.



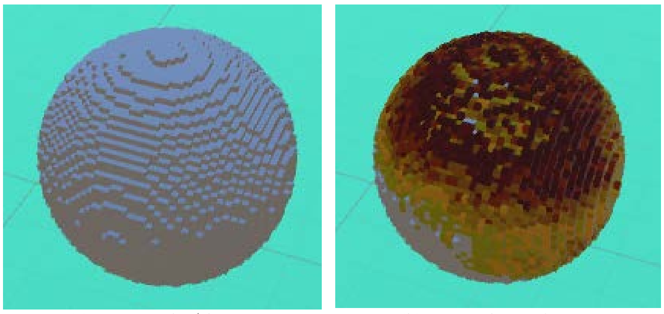
(a) Executed 1st loop (b) Iterated 200 times

Fig. 3. Horizontal iron plate



(a) Executed 1st loop (b) Iterated 200 times

Fig. 4. Slanted square pipe



(a) Executed 1st loop (b) Iterated 200 times

Fig. 5. An iron ball

B. Comparison of rust propagation due to tilt

It is possible to determine the direction in which rust propagates easily due to the inclination of a 3D object. Comparison results of rust propagation with different tilt angles are shown in Fig. 6. In the comparative studies, the same iron plate as in Fig. 3 was used.

Fig.6 (a), Fig.6 (b), and Fig.6 (c) show the rust propagation results of inclination angles of 0°, 45°, and 90°, respectively, after 150 times iteration.

Regarding Fig. 6 (b) and Fig. 6 (c), the upper and lower sides of the figures correspond to the upper and lower sides of the iron plate. It is easy to propagate downward due to the inclination angle.

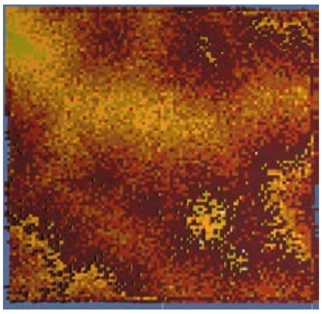
In addition, depending on the situation, the tilt angle of an object existing in real space may change during the rust progress. Fig. 7 shows the results when the tilt angle changes during rust progression. The object shown in Fig. 7 is similar to one in Fig. 3.

Fig. 7 (a) shows the object is, first placed in horizontal (0°) and iterated 100 times of the rust propagation. Then, the object's tilt angle is changed to vertical (90°) and 50 times of the rust propagation are executed.

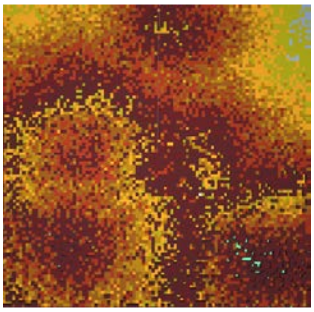
Fig. 7 (b) is vice versa. First, 100 times iteration is executed where the object is placed in vertical (90°). Then, additional 50 times of the rust propagation are executed to the object is placed in horizontal (0°).

Note that the initial rust is generated only from the center of the object.

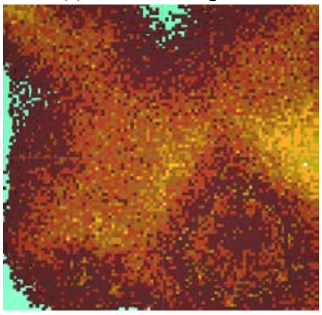
As shown in Fig.7, generated rust texture images are greatly different between each other.



(a) Inclination angle 0°

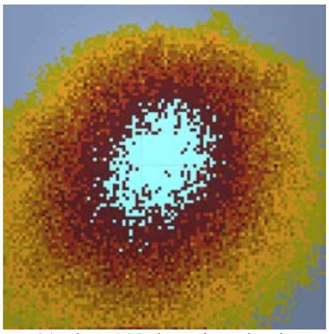


(b) Inclination angle 45°

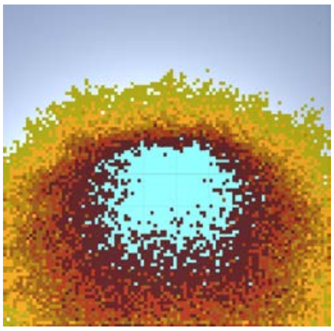


(c) Inclination angle 90°

Fig. 6. Comparison of rust propagation due to tilt after 150 times iteration



(a) First, 100 times iteration in horizontal, then, additional 50 times iteration in vertical



(b) First, 100 times iteration in vertical, then, additional 50 times iteration in horizontal

Fig. 7. The midway change of the tilt angle

C. Comparative evaluation with real rust

Fig. 8 shows a photograph of the real rusted iron plate. The real iron plate was placed in horizontal. Comparing Fig. 8 and Fig. 6 (a), our proposed method can generate realistic rust texture images.

Fig.8. The real rusted iron plate

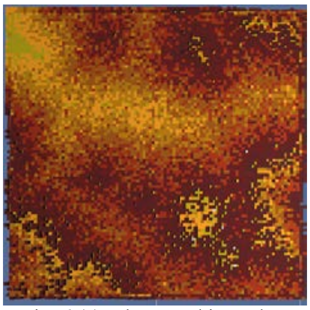


Fig. 6 (a). The rusted iron plate

From these results described above, it can be concluded that our proposed method is very effective and efficient to generate rusted images.

V. Conclusions

In this paper, we proposed a voxel automaton method for representing rust progression to simulate the processes of iron rusting and eventually shape deformation due to corrosion. By using our proposed method, it became possible to reproduce not only discoloration due to corrosion and irregularities, but also how holes are opened on an iron plate by the progress of rust.

For future work, we would like to express irregularities on iron plate surfaces considering the accumulation of rust separated by corrosion. It might be possible to reproduce the unevenness of surfaces of a more corroded object by expressing the deposit.

Acknowledgement

We would like to thank all the members of Visual Computing Lab., Tokyo Denki University for their encouragement and fruitful discussion.

This work was supported by JSPS KAKENHI Grant Numbers JP-17K19965 and JP-17H004436.

References

[1] Yao-Xun Chang, and Zen-Chang Shih, "The synthesis of rust in seawater", The Visual Computer, Vol.19, No.1, pp.50-60, 2003.

[2] Y. Kamata, Y. Manabe, and N. Yata, "Automatic generation of physical erosion map based on the object shape", ITE Winter Conference Proceedings, Article 2-2 (2 pages), 2012. (In Japanese)

[3] T. Doi, N. Kato, J. Kim, "CG simulation of rust growth of metal products with time", Abstract of Graduation Thesis, Department of Information Systems and Mathematical Sciences, Nanzan University (4 pages), 2009. (In Japanese)

[4] R. Tanabe, T. Moriya, T. Takahashi, "A generation method of rust aging texture considering rust spreading", In Proceedings of International Workshop on Advanced Image Technology (IWAIT) 2015, Paper ID 399 (4pages), 2015.

[5] K. Kanazawa, R. Tanabe, T. Moriya, T. Takahashi, "Visual Simulation for Rust Aging Considering Object's Geometry", Journal of The Institute of Image Electronics Engineers of Japan, Vol.46, No.4, pp.547-558, 2017. (in Japanese)