

# 頂点情報と UV を用いたウェザリングシミュレーションの研究

濱川 史都<sup>a)</sup>・坂本 真人<sup>b)</sup>

## Study of Weathering Simulation Using Vertex Information and UVs

Fumito HAMAKAWA, Makoto SAKAMOTO

### Abstract

In recent years, three-dimensional computer graphics (3DCG) technology has been developed. In addition, much research has been done on weathering representations such as rust and moss for realistic representation. However, when outputting simulation results on an image, a large number of images are required to display different simulation results in a three-dimensional space. In this paper, a simulation method using vertex information of a 3D model and simple images is proposed. In this method, when the number of vertices is sufficient, the simulation result output does not use images, thus reducing the data increase. This figure is one of the experimental results

**Keywords** : 3DCG, Rust, UV, Vertex, Weathering,

## 1. 研究背景

近年、3次元コンピュータグラフィックス(以下3DCG)は、多くの分野で利用されている。その用途は、映画やゲームのようなエンターテインメント、VRやARを用いた教育分野、医療や建築分野等多岐にわたる。CGは環境や時間のような干渉を受けることがないため、現実のように劣化することはない。よって現実で物体が劣化するウェザリングをCG中で表現する研究が多く存在する。

土井ら[2009年]は、セルラオートマトンを用いて金属製品における時間経過での錆成長のシミュレーションを行った<sup>1)</sup>。手法は、2次元セルラオートマトンを用い、それぞれのセルとセルの近傍を対象に錆の発生、進行、成長をシミュレーションした。

田邊ら[2014年]は、本錆と副錆の挙動を考慮したウェザリングシミュレーションを行った<sup>2)</sup>。手法は、土井ら[2009年]同様に、セルラオートマトンを用いた錆の発生、進行のシミュレーションである。この研究では、本錆、副錆それぞれの錆びやすさや成長レベルを用い、傾きに応じた成長範囲のパラメータで錆の進行を制御することで本錆と副錆が重力、傾きを用いて伝播する様子を表現した。

鈴木ら[2017年]らは、3次元オートマトンを用いて鉄が腐食するシミュレーションを行った<sup>3)</sup>。この研究では、3DCGで作成した金属製品に対し、現実の金属製品にお

ける錆が発生、成長することによる穴が開くような表現を可能にすることを目的としている。

しかし、これらの多くはシミュレーション結果を画像に出力することやUVを調整する必要があり、ベースカラーが指定されているように既存のモデルに適用しづらい問題点も存在する。これらの問題点を解決するため本研究では3DCGの隣接する頂点の情報とUVを用いたシミュレーションを提案した。

## 2. 実行環境

本研究のシミュレーションは表1の環境で実行を行った。異なるソフトウェアにおいても実行可能か検証するために本稿では2つのソフトウェアを用いている。

表1. 実行環境.

OS	Windows 10
実行ソフト1	Unity
実行ソフト2	Blender
スクリプトツール	Visual Studio
デザインツール	Adobe Illustrator 2023
プログラム言語	C#
	Python

a) 工学専攻機械・情報系コース大学院生

b) 工学基礎教育センター教授

### 3. 頂点の隣接リスト

本研究におけるウェザリングシミュレーションは、錆や苔が伝播する様子を接続している頂点同士の局所的関数で表現するため、シミュレーション対象の 3D モデルのポリゴンが三角面、四角面またはその両方を用いても問題なく実行することができる。これを実現するための頂点の隣接情報を取得し格納する隣接リストの実装方法を本章にて示す。

#### 3.1. Blender における隣接リスト作成

Blender ではスクリプトとして Python がサポートされており、「bmesh」ライブラリを用いることでメッシュ情報を取得することができる。エッジは頂点同士を結んだ辺であるため、エッジ  $e_0$  を構成している 2 頂点  $v_0$  と  $v_1$  は隣接しているためメッシュを構成するすべてのエッジを探索することでメッシュに含まれるすべての頂点の隣接情報を取得することができる(図 11)。

#### 3.2. Unity における隣接リスト作成

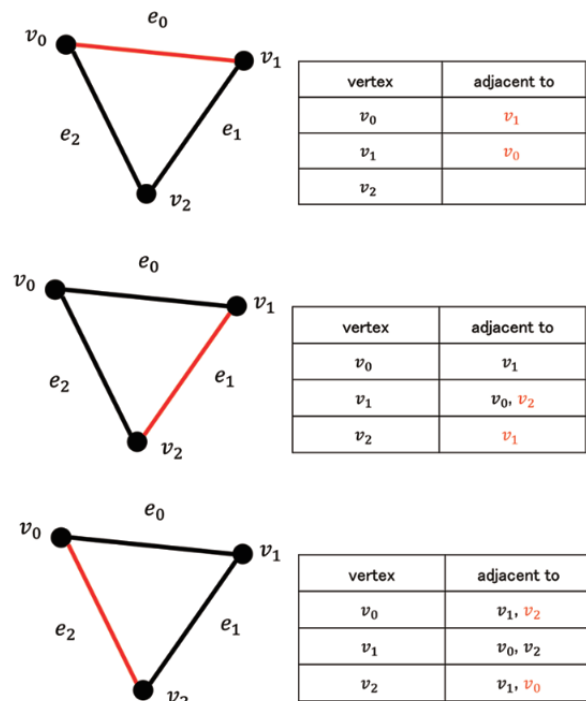


図 1. Blender における頂点リストの例.

Unity ではスクリプトとして C# がサポートされており、「Unity.Engine.Mesh」を用いることでメッシュ情報を取得することができる。Unity.Engine.Mesh のクラスである Mesh.triangles を用いてメッシュの三角面ポリゴン情報を取得する。すべてのポリゴンが三角面である場合、ポリゴン  $p_0$  を構成する頂点  $v_0$ 、 $v_1$ 、 $v_2$  はそれぞれ隣接しているため、ポリゴンを構成する頂点を取得することで頂点の隣接情報を取得することができる。Mesh.triangles は、メッシュオブジェクトに含まれる三角面ポリ

ゴンを順に構成する頂点のインデックスとその頂点の順番を格納する一次元配列である。すべてのポリゴンに対して構成する3頂点のインデックス番号を参照し、隣接情報をリストに格納する(図 2)。

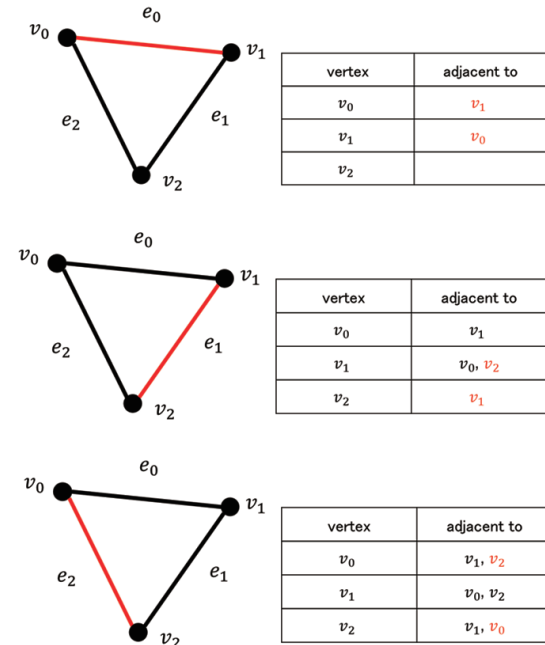


図 2. Unity における頂点リストの例.

### 4. シミュレーション手法

本研究での錆のウェザリング表現は、錆が各頂点においてランダムで発生し、重力方向に伝播しやすいように頂点の三次元座標を用いて錆が発生、伝播するかを決定している。本研究での錆の伝播規則は以下のように決定される。

- 1 錆の進捗最大値を決定する (本稿は 1000)
- 2 全頂点に対して以下を行う
  - 2.1 探索している頂点に隣接する全頂点に対して以下を行う
  - 2.2 乱数を生成
  - 2.3 乱数を含まない遷移関数の値が、2.2 の乱数を越えている場合進捗を+1
  - 2.4 10 万分の 1 の確率で錆の進捗が+1
- 3 錆の震度を頂点の UV 情報に反映

### 5. 実験内容

本稿の実験ではシミュレーション UV を図 2 左のマスクテクスチャと図 2 右のカラーテクスチャに適用している。

マスクテクスチャは、グレースケール画像であり、シミュレーション結果をソフトウェアのシェーダーに入力することで 3DCG モデルの表面の金属感や粗さのような質

感を表現する際に活用することができる。そのほかにも、凹凸のある錆びている表面を表現するためのノイズテクスチャや既存の 3DCG モデルのベースカラーの適用範囲を制御に用いている。

カラーテクスチャは、カラースケールを表す。また、カラースケール用テクスチャにはアルファ値を設定し、一定の値以上で剥離し、穴が開く表現を実装した。

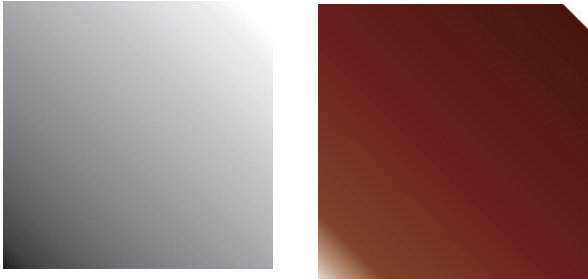


図 2. 用いたテクスチャ.

本稿ではシミュレーションの特徴を示すため実験を 4 種類行っている。

6.1 の基礎実験では、立方体モデルを x 軸中心に回転させ、傾けている。このシミュレーションでは、錆が重力方向を考慮して伝播するように確率を用いた遷移規則で隣接する頂点の浸食度が増加する。本研究の遷移規則は、隣接する頂点同士での 3 次元座標を用いた高さによって遷移のしやすさが異なり、高さの差異が大きいかほど自身より低い座標の頂点へは遷移しやすく、自身より高い座標の頂点に対しては、遷移しにくい。図 4 の場合、y 軸を高さとし、y 軸正の方向にいくほど高さが高くなるとする頂点の高さ  $h$  は

$$h(v_2) < h(v_0) = h(v_1) = h(v_3) < h(v_4)$$

である。ここで自身より高い位置の頂点に対しての遷移関数の値は、自身より低い位置の頂点に対しての遷移関数に対して低くなる。よって  $v_0$  からの遷移関数  $T(v, v_0)$  は、

$$T(v_4, v_0) < T(v_1, v_0) = T(v_3, v_0) < T(v_2, v_0)$$

となる。同じ規則で 6.2、6.3、6.4 に対しても実験を行った。

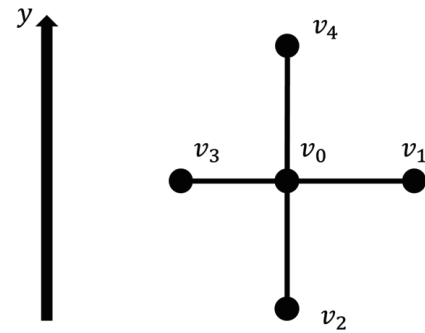


図 4. 高さが異なる頂点における遷移のしやすさ.

## 6. 実験結果

### 6.1. 基礎実験結果

マスクテクスチャをベースカラーとして用いたシミュレーション結果は、図 5 左であり錆が進行している部分は白く、進行していない部分は黒い。立方体の上部は錆が発生しておらず、伝播もしづらいため浸食していないことがわかる。また、立方体の下部は錆が伝播しやすいため白く、中部は対応する頂点より上部に錆が発生しているかで錆の浸食度が変化していることから、錆が下部に伝播していく様子を表している。

カラーテクスチャをベースカラーとして用いたシミュレーション結果は、図 5 右においてマスクテクスチャで黒く浸食度が低い部分は、立方体の白い色であり、マスクテクスチャで白い錆の浸食度が高い部分は錆の色、更に浸食度が高い部分は、カラーテクスチャのアルファ値を参照し、透明であるため剥離を表現している。

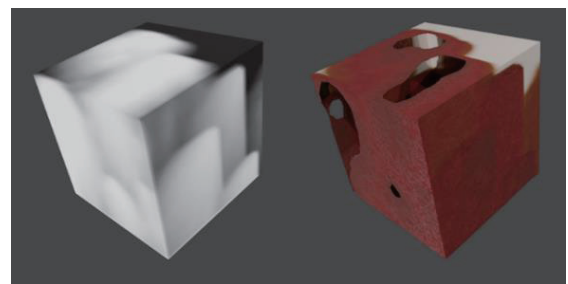


図 5. 傾きのある立方体に対しての実験結果.

### 6.2. 乱数実験結果

マスク結果は、エラー! 参照元が見つかりません。左で、カラー結果は、エラー! 参照元が見つかりません。6 右である。乱数により 6.1 の結果と違い、錆の発生、伝播が異なるため、立方体表面の色、剥離部分が違うことから複数のパターンを出力することが実証できている。

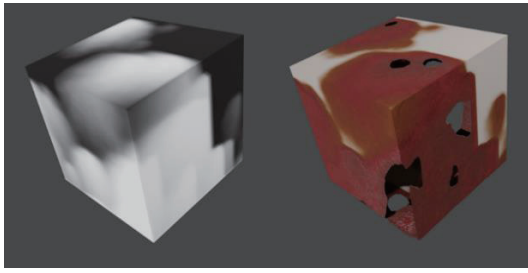


図 6. 確率における複数パターン生成検証.

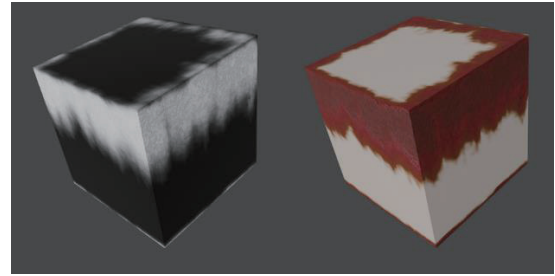


図 8. 頂点法線を用いた錆の発生場所制御結果.

### 6.3. 形状変化実験結果

本実験では、錆が重力方向に伝播することを表現するために隣接する頂点同士の 3 次元座標を参照し、錆の伝播関数に用いている。また、頂点の 3 次元座標を初期状態で記憶せず、毎ステップの伝播関数適用時に頂点の 3 次元座標を参照することでステップ間にオブジェクトが回転した場合でも対応することができる。

マスク結果は、図 7 左で、カラー結果は、図 7 右である。結果は、いずれも 6.1、6.2 と違い右手前方向にオブジェクトが傾いているため、錆が傾きに沿って伝播していることがわかる。

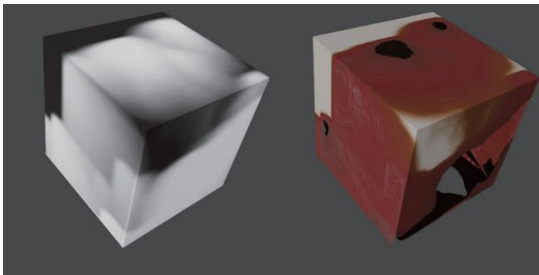


図 7. 傾きによる錆の伝播方向の違いの検証.

### 6.4. 発生箇所制限実験結果

本実験では、頂点情報の 1 つである法線情報を用いる。頂点自身の法線と隣接する頂点の法線を用いることで自身がメッシュオブジェクトの角に存在するかどうかを知ることができる。立方体の場合、隣接する頂点との法線の x 軸成分と z 軸成分の差が一定の値以上、また y 軸成分と z 軸成分の差が一定の値以上の頂点のみを錆が発生する場所と設定することで立方体の上面、下面のエッジから金属が錆びる様子を表現することができる。

結果は図 88 であり、以前までの結果と異なり錆の発生部分が上面と下面のエッジから発生し、そこから傾きに従って伝播しているため上面の中心や立方体の高さが低い部分には錆が伝播していない。

## 7. 考察

6.1 の実験では、錆が発生した部分からオブジェクトの傾きを考慮して伝播していくことが確認できた。また、PNG 画像におけるアルファ値を用いることで 3D モデルに対して錆の進捗が高い部分に穴が開くような表現を実装することができた。

6.2 の実験結果では、6.1 と同じ条件でシミュレーションを行うことで錆の発生、進行におけるランダム性を検証した。2 つの実験で灯籠や街灯のように同じ 3DCG を並べて配置するような場合に個別にシミュレーションを行うことにより個別にテクスチャを増やさず、異なるパターンの結果を出力できることを示せた。

6.3 の実験結果では、6.1 と傾きを変えた状態の 3D モデルに対してシミュレーションを行った。6.1 で x 軸中心に回転させたものを 6.3 で y 軸中心に回転させたことでオブジェクト全体の傾きを変えたが、錆が傾き方向に伝播していくことが示せた。これにより、頂点の隣接情報のみでも全体の変化を考慮して実行できることを示せた。

6.4 の実験結果では、6.3 までの錆の発生場所が全体に一葉であったのに対し、頂点の 3 次元座標だけでなく法線情報を用いることでオブジェクトのエッジ部分を認識し、錆の発生部分を制御できることを示せた。

## 8. まとめ

本研究では、3DCG の頂点情報と UV を用いたウェザリングシミュレーションを行った。先行研究と比べ既存のモデルに適用しやすくオブジェクト個別にテクスチャを増やす必要がないという利点がある。しかし、より複雑な模様を表現する場合は、頂点数を増やす必要があるため、描画に負荷がかかる可能性がある。頂点数が増えることによる描画の負荷を抑えるためには、シミュレーションを行うオブジェクトを制限する、複数のパターンや結果のみが必要であれば先行研究同様テクスチャに結果をバイクすることなどが有効と考える。これからは、本研究で行わなかった種類の違う錆や苔等の種類に応じたシミュレーションも進めていく必要がある。

## 参考文献

- 1) 土井 崇義、加藤 直大、金 知俊: “時間経過に伴う金属製品の錆成長の CG シミュレーションに関する研究” [Online].  
<http://www.st.nanzan-u.ac.jp/info/grthesis/it/proc/2009/06mi009.pdf>
- 2) 田邊 竜馬、森谷 友昭、森本 有紀、高橋 時市郎:  
“本錆と副錆の挙動を考慮した錆の経年変化テクスチャ生成”, ITE Technical Report (in Japanese), Vol.38, No.16, pp.95-98, 2014.
- 3) 鈴木雄太、森谷友昭、高橋時市郎、高橋時市郎:  
“3D オートマトンを用いた鉄の腐食による形状変化”,  
2017 年映像情報メディア学会年次大会 (in Japanese), 21C-2, 2 pages, 2017.

