

# 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

2006MI009 土井 崇義 2006MI063 加藤 直大  
指導教員 金 知俊

## 1 はじめに

近年、コンピュータの性能の向上により、コンピュータグラフィックス(CG)は実写とほとんど区別がつかないほどリアルな映像を作れることができるようになってきている。また様々なマッピング法などのCG技術の向上により、完璧でクリーンな外観の物体をCGにより容易に表現出来るようになった。しかし、完璧で劣化のない画像はかえって現実感を損ねているということもある。現実の物体でそこまでクリーンな物体は存在しない。例えば人々の生活に密着した日常生活用品である食器やガラスなどには、何かしら汚れ、傷、錆などの生活空間における表面劣化が起こっているはずである。またアニメーションや映画などでCGを使う際も、その風合いを表現する錆は現実感を出すために大変重要である。以上より、我々はより現実的でリアルな金属製品の表現には、汚れ、傷、錆などの劣化をCGで表現することが必要だと考え、その中でも、自然環境で劣化をもたらす金属製品の錆に着目し、そのための手法を研究した。なお、加藤直大は主にプログラミングの部分を、土井崇義は主に錆のしくみ等についてを担当した。

## 2 関連研究

原田英一はCGを使った錆の表現方法について研究した[10]。この研究は鉄錆の進行を様々な環境下でシミュレートすることを目的としている。海沿い地域や都市部などの環境による錆の色の違い、接している水の量による錆の進行速度の違い、などを表現するために「時間・水の量・錆進行速度・環境・色」のパラメータをとった。プログラムは8近傍のセルを作り、錆を配置する。そこにパラメータの情報を与え錆を成長させることでシミュレーションを行った。さらにより現実感を出すため、錆の色は実際にその環境下で発生した錆を撮影し、近似させた。

またYao-Xun ChangとZen-Chung Shihは海水中における鉄製品の錆について研究した[8]。この研究は文献[10]と同様に、海水中での鉄錆の進行をシミュレートすることを目的としているが、表現手法にLシステムを使用している。Lシステムとは複雑な分岐構造を記述することが出来るため、CGにおいては木や草などの植物を作る際に使われるツールである。錆の分岐構造をLシステムで記述することで、錆の成長をより自然に表現出来る様にした。本研究は文献[10]の8近傍セルを使うプログラムを参考とし、そこに独自のパラメータを使うことでシミュレーションを行った。

## 3 錆について

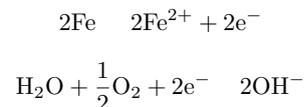
錆とは金属の腐食生成物のことを言う。また専門的分野である腐食防食工学では、鉄や鉄合金の腐食生成物のみを「錆」と呼び、鉄以外の金属では錆とは呼ばずに単に腐食生成物と呼ばれている[1]。

### 3.1 錆発生原理

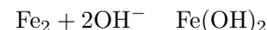
錆は酸化還元反応により物質の表面が電子を失ってイオン化し、表面から離れて行くことで進行する。これは電気化学的な反応なので、錆が発生するかどうかは電位とpHに依存する。生じたイオンは酸素により酸化物、または水により含水酸化物に変化して物質表面に堆積する。酸素により腐食することを乾食、水により腐食することを湿食と言い、このように腐食が起こるには水と高い温度が必要である[1]。

乾食の代表的なものは酸化であり、鉄は大気中の酸素によって表面に酸化物をつくる。しかしこの酸化物は極めて薄く、透明であるため、錆という意味では全く問題にならない。温度が200を超えると黄褐色、赤紫色、青色などの色が着いていくが、自然界で影響が見られるのは高温のガスに触れた時くらいである。

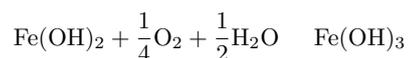
また水を必要とする湿食とは、表面に形成された水の層によりイオンとなって溶け出すのが原因である。鉄を例として調べた[2]。



これより



が得られる。この水酸化鉄(Fe(OH)<sub>2</sub>)が錆となり、さらに出発物質として表面積を増やすため、一度発生した錆は加速度的に進行する。また時間が経つと水酸化鉄はさらに酸化し



となる。ここで発生したFe(OH)<sub>3</sub>は赤褐色の赤さびである[2]。同一の金属から発生した錆にもFe(OH)<sub>2</sub>やFe(OH)<sub>3</sub>のように異なる性質の錆となる場合もある。

### 3.2 錆の腐食速度

錆の腐食速度は様々な要素によって異なる。外部からの環境だと海水は錆やすかったり、ステンレス保護してある金属は錆にくい、などこれらは接している水の量やpHなどが影響している。他にも錆が発生する金属の種

類によっても腐食速度は異なってくる。本研究では鉄以外に銅も対象としているので、この2つの金属の腐食速度を調べた。金属の表面に錆が出来るとその後の錆の生成になんらかの影響を与える。錆の生成は図1に示したように4から5年経過した後、安定期に入るがこの時期に銅や鉛のような自然化成皮膜型金属や、それらを含む鉄合金である耐候性鋼のような錆びなくなるグループと、一般的な鉄鋼である普通鋼のように錆が進行するグループに分かれる [3]。

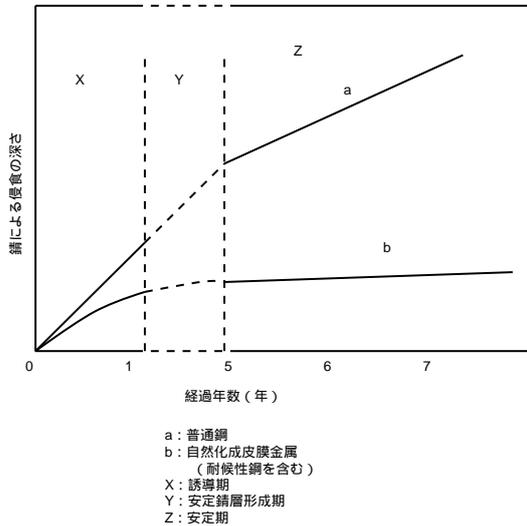


図1 錆層成長の模式図

このように錆の進行速度は様々な要因によって変化するため、現実の錆の進行はまだ完全に予測出来ない。しかし錆の発生する金属の性質や環境、周りとの相対性を考えることでより現実に近づけることは可能である [3]。

#### 4 セルオートマトン法

今回、錆生成のシミュレーションを行う手法としてセルオートマトン法を用いている。セルオートマトン法とは、平面上に格子状のセルを敷き詰め、各セルごとに有限種類の離散的内部状態量を定義し、その離散的内部状態量を近傍のセルとの相互作用(局所近傍則)のみを考慮することで離散時間ステップごとに状態転移則に従い推移させ、全体としての現象を表現させる方法である [4][5]。

この手法は離散的計算モデルの中でも手順、規則が非常に単純化されていて、また、複雑な自然現象を表すのに適しているため、本研究で用いている。

#### 5 シミュレーションの手法

本研究のシミュレーションの手法は、OpenGL と呼ばれるプログラミング・インターフェースを使用し、平面上の金属と仮定したテクスチャに錆を発生させ、セルオートマトン法を用いて錆の成長のシミュレーションを

行い、作成されたテクスチャを金属製品を仮定した形状データにマッピングする。

##### 5.1 OpenGL について

OpenGL とはハードウェアや OS などに依存しないグラフィックスライブラリ (API, Application Programming Interface) である。これは 2D・3D グラフィックスを描くためのプログラミング・インターフェースであるが、2D の表示よりゲームなどの 3D-CG 映像のリアルタイム生成に高い性能と機能を発揮する。またオープン仕様として公開され幅広い OS に対応していることや、豊富な補助ライブラリがあることなどから広く一般に普及しており、事実上の 3D グラフィックス・ライブラリの標準となっている [6]。

また今回は GLUT (The OpenGL Utility Toolkit) という OpenGL に基づいて設計された、C 言語のライブラリも使用している。これを使うことでマウスやキーボードからの入力によってウィンドウを操作したり、高品位な立方体や球体として再現することも可能である。この様に OpenGL に GLUT を組み合わせる事で

- UNIX 系 OS と Windows と Mac のいずれでも動く
- リアルタイムに 3D-CG を表示出来る
- プログラムがとても簡単に書ける

というメリットが得られる [6]。

##### 5.2 錆テクスチャの作成

###### 5.2.1 錆テクスチャ作成の流れ

本研究では次のような流れで錆テクスチャ作成のシミュレーションを行う。

- 錆の発生
- 錆の進行
- 錆の成長

錆の発生は、二次元平面上に錆を発生させるステップ、錆の進行は発生させた錆が拡大していく挙動を表現するステップ、錆の成長は錆の進行度合いが進んでいくステップとなっていて、本研究ではこの3つの段階を経て進める。

###### 5.2.2 初期設定とセルのパラメータ

本研究では、まず二次元平面上に 256 × 256 の格子状のセルを敷き詰め、各セルごとに離散的内部状態量(パラメータ)を設定し、パラメータの変化によって錆が進行していくというメカニズムで錆を表現した。各セルが持つパラメータは以下ようになる。

- セルの各頂点座標 (x,y)
- セルの表面属性の情報 (Diffuse, Ambient, Specular)
- セルの錆の成長レベル (正の整数)
- セルの錆の段階レベル (1,2,3,4)
- セルの錆やすさの情報 (0:通常の状態, 1:錆やすい状態)

各頂点座標はセルの頂点の座標、表面属性の情報は各セルごとの表面の属性を Diffuse(拡散反射成分)、Ambient(環境光反射成分)、Specular(鏡面反射成分)の3つの成分を配列として格納している。また、錆の成長レベルは各セルの錆の進行度を正の整数で表していて、錆の段階レベルは錆がどの段階まで錆びているかを表すパラメータである。錆やすさの情報パラメータは初期段階で錆を発生させるときに、あらかじめ任意に与えておくことにより、錆びる確率が高くなるというパラメータで、局所的な錆の発生などに使用する。

### 5.2.3 二次元錆進行セルオートマトン

本研究では、錆進行の挙動を表現するためにセルオートマトン法のしくみを利用し、二次元錆進行セルオートマトン法を考案した。そのルールを以下に示す。図2、3はルールをわかりやすくしたものである。

- セルの現在の成長レベルが1以上の時、次のステップ時には周囲のセルのうち一つのセルの成長レベルが+1される。
- セルの現在の成長レベルが一定の値になったとき、段階レベルが+1される。

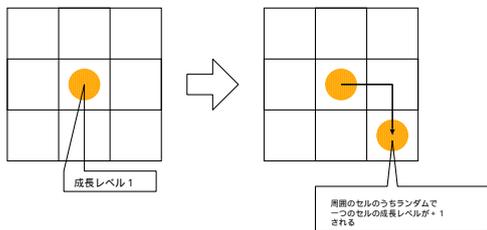


図2 二次元錆進行セルオートマトン法のルール1

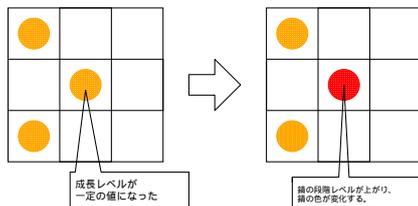


図3 二次元錆進行セルオートマトン法のルール2

### 5.2.4 錆の発生

本研究では錆の発生には乱数を用いている。二次元平面上の全てのセルは通常状態で1ステップごとに0.1%の確率で錆が発生し、錆の成長レベルが+1される。錆の成長レベルが1になると段階レベルが+1され錆が成長する。段階レベルが1になると表面属性が変化し、錆色となる。また、発生の際にあらかじめ錆やすさのパラメータを与えておくことにより局所的な錆の発生や、文字型などの任意な形で錆を発生させることができる。錆やすさのパラメータを与えられたセルは1ステップごと

の発生確率が0.1%から10%まで上がり、錆が発生しやすくなる。



図4 ランダムな発生



図5 局所的な発生

### 5.2.5 錆の進行

段階レベルが1となり錆として認識されたセルは次の時間ステップには隣接したセルに進行する。錆の進行は、錆として認識されたセルの周囲8マスのセルのうち、ランダムに1つのセルの成長レベルが+1され錆状態となる。

### 5.2.6 錆の成長

錆の発生、移行が進んでいき、あるセルの成長レベルが一定の値に達したとき、そのセルの錆は成長する。着目したセルの段階レベルが上がるとその値に対応した表面属性の色に変化する。錆の成長レベルと段階レベル、表面属性のデータの関係図を図6に示す。

段階レベル	成長レベル	色
1	1	
2	3	
3	8	
4	10	

図6 成長レベルと段階レベルと表面属性の関係図

## 6 シミュレーション結果

### 6.1 マッピングの手法

本研究では、考案した手法により作成した画像データはテクスチャとして変換し、物体表面にマッピングすることにより高度なレンダリングを表現している。マッピングの手法としては、左の画面で進行している錆の画像データをテクスチャとして変換し、動的に右側の金属製品データに張りつけて表示することにより、金属平面上で錆が進行していく表現が可能とした。今回は金属製品の中でも代表的に使用されているティーポットの形状

データを使用してシミュレーションを実行する。

## 6.2 シミュレーションの実行

実際にシミュレーションした画像データを図 7 から図 10 に示す。画像左側が生成したテクスチャで、右側がテクスチャを形状データにマッピングした画像である。

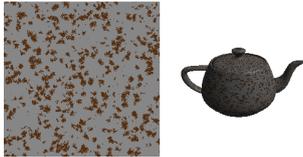


図 7 10 ステップ後のテクスチャとマッピングしたティーポットの画像

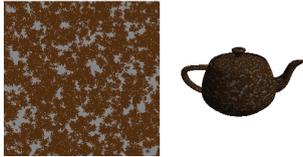


図 8 20 ステップ後のテクスチャとマッピングしたティーポットの画像



図 9 30 ステップ後のテクスチャとマッピングしたティーポットの画像



図 10 50 ステップ後のテクスチャとマッピングしたティーポットの画像

実際にマッピングしてみてティーポットが錆びていく様子を確認することができた。

## 6.3 銅の錆

また、本研究で考案した手法を利用して金属、錆の色の表面属性を変化させることにより鉄だけでなく、銅の錆（緑青）の表現も可能とした。金属を銅に変更してシミュレーションした結果を図 11 に示す。



図 11 緑青錆のテクスチャとティーポット

## 7 終わりに

本研究はよりリアルな金属製品を CG で表現するために金属の錆に着目し、鉄、銅の 2 種類の金属の錆生成シミュレーションの手法を考案した。手法としてはセルオートマトン法を利用して二次元平面上にセルを敷き詰め、表面属性を設定し金属テクスチャとして表現させ、金属のテクスチャ上に錆を発生させ進行させていく。錆の発生はランダムで、時間ステップを進めていくと発生した箇所からは錆は進行、発生していない箇所からは乱数により新たに錆の発生を繰り返す。シミュレーション時には作成したテクスチャを動的にティーポットの形状データにマッピングしていくことによって時間ステップごとにティーポットの表面上での錆の発生の再現も可能にした。この手法を用いてシミュレーションを実行した結果、現実と類似した錆が進行した金属製品を作ることができた。改善すべき点としてはテクスチャ上のデータに法線ベクトルを設定し、バンプマッピングを用いることや、錆の進行に伴う金属の表面劣化や孔食と呼ばれる金属に穴が空く現象の表現、金属の種類や周囲の環境による錆の進行速度の変化、錆の色、錆の広がり割合などの実画像との数値な検証などが今後の課題として挙げられる。

## 参考文献

- [1] 松島巖：“トコトンやさしい錆の本”，シナノ 2002.
- [2] 松島巖：“錆と防食のはなし”，日刊工業新聞社 1993.
- [3] 増子 昇：“さびのおはなし”，日本規格協会 1990.
- [4] 林武文、加藤清敬：“3 次元 CG プログラミング”，コロナ社 2008.
- [5] 森下信：“セルオートマトン 複雑系の具象化”，養賢堂 2003.
- [6] 加藤恭義、光成友孝、築山洋：“セルオートマトン法”，森北出版株式会社 2002.
- [7] 床井浩平：“GLUT による OpenGL 入門”，工学社 2008.
- [8] Yao-Xun Chang and Zen-Chung Shih：“The synthesis of rust in seawater”，The Visual Computer, Vol.19 No.1, pp.50-66 (2003.3).
- [9] Stephane Gobron, Denis Finck, Philippe Even, Bertrand Kerautret：“Merging Cellular Automata for Simulating Surface Effects”，Lecture Notes in Computer Science, Vol.4173/2006, pp.94-103 (2006.6).
- [10] 原田英一：CG による錆の表現，<http://verygood.kyushu-id.ac.jp/~harada/GS/s1d001.html>.