

2012年度 卒業論文

3DCG モデリングにおける  
古紙の劣化表現の制作支援に関する研究

指導教員：渡辺 大地 講師

メディア学部 ゲームサイエンスプロジェクト  
学籍番号 M0109043  
和泉 真

2012年度 卒業論文概要

論文題目

3DCG モデリングにおける  
古紙の劣化表現の制作支援に関する研究

メディア学部

学籍番号：M0109043

氏名

和泉 真

指導  
教員

渡辺 大地 講師

キーワード

モデリング、紙、劣化、  
制作支援、ツール

近年、3DCG コンテンツの開発が盛んに行われている。3DCG コンテンツにおいて、登場するモデルをより実物らしく見せることが必要となる。そのために実物に備わっている性質をモデルにおいて再現することが重要である。本研究では、物が持つ性質の中でも劣化という現象に注目した。劣化表現は、年月の経過によって古びた状態を表現するために重要な要素であり、これを取り入れた映像作品も見られるようになった。しかし、劣化を表現するにあたって行わなければならない制作工程は手間や時間がかかるものが多く、制作においてコストがかかる。紙の劣化においては、劣化の要素として切れ、破れ、折れ、湾曲、変色が挙げられる。この中でも折れの表現はテクスチャに描画することで表現しようとした場合に、様々な視点からモデルを見る可能性の高い3DCG コンテンツ上では意図しない出力結果になる問題が発生する。一方で、ポリゴンの変形による制作では手間や作業時間がかかるという問題点がある。既存研究では、よりリアルな劣化表現を目指す研究は数多くあるが、制作効率化を対象とした研究はない。そこで本研究では、紙の劣化を表現する上での破れ目、切れ目を表現する工程を自動で形状変化を行う手法を提案する。紙の切れ目の幅や長さ、切れ目の入る角度のノイズの大きさなどの様々なパラメータを任意の値に設定することで、多様な切れ目、破れ目の劣化を表現する紙モデルを生成する。この手法の妥当性を検討、考察し、手法の有効性を示した。

# 目次

<b>第1章</b>	<b>はじめに</b>	<b>1</b>
1.1	研究背景	1
1.2	論文構成	6
<b>第2章</b>	<b>切れ目、破れ目の生成の提案手法</b>	<b>7</b>
2.1	劣化表現生成手法の概要	7
2.2	劣化表現生成手法の流れ	7
2.3	手法で用いるパラメータについて	8
2.4	基本図形の生成	9
2.5	切れ目の生成	11
2.5.1	切れ目の開始地点の決定	11
2.5.2	切れ目の幅の生成	12
2.5.3	切れ目の角度の決定	13
2.5.4	切れ目頂点の移動量の決定と切れ目の生成	14
2.5.5	切れ目の伸長	14
2.6	破れ目の生成	17
2.7	破れ目からの切れ目の生成	18
<b>第3章</b>	<b>動作検証</b>	<b>20</b>
3.1	実装	20
3.2	ツール概要	20
3.3	評価	22
3.4	問題点	31
<b>第4章</b>	<b>まとめ</b>	<b>33</b>
	<b>謝辞</b>	<b>34</b>
	<b>参考文献</b>	<b>35</b>

# 目次

1.1	コンテンツ用に制作された古紙テクスチャの一例 . . . . .	2
1.2	劣化が確認できる古紙の一例 . . . . .	3
2.1	手法のフローチャート . . . . .	8
2.2	4点で構成した正方形 . . . . .	10
2.3	頂点を新たに生成した図形 . . . . .	11
2.4	切れ目の幅を作るために新たに頂点を生成 . . . . .	12
2.5	切れ目の幅の決定 . . . . .	13
2.6	頂点の移動による切れ目の生成 . . . . .	14
2.7	内分比手法による切れ目 . . . . .	15
2.8	移動量手法による切れ目 . . . . .	15
2.9	切れ目の伸長用の頂点を生成 . . . . .	16
2.10	切れ目の伸長 . . . . .	17
2.11	$e$ の値が $d$ の値以下の時に 頂点を辺で結ぶ . . . . .	18
2.12	辺を結ぶことで新たにできた面を削除し破れ目を生成 . . . . .	18
2.13	破れ目に新たな頂点を生成 . . . . .	19
2.14	生成した頂点から切れ目を生成 . . . . .	19
3.1	紙の劣化自動生成ツール . . . . .	21
3.2	$p = 0.1$ の時の出力結果 . . . . .	22
3.3	$p = 0.3$ の時の出力結果 . . . . .	22
3.4	$p = 1.0$ の時の出力結果 . . . . .	23
3.5	$m = 2.0$ 、 $n = 1.0$ の時の出力結果 . . . . .	24
3.6	$m = 20.0$ 、 $n = 19.0$ の時の出力結果 . . . . .	24
3.7	$m = 20.0$ 、 $n = 1.0$ の時の出力結果 . . . . .	24
3.8	$z = 1.0$ の時の出力結果 . . . . .	25
3.9	$z = 4.0$ の時の出力結果 . . . . .	25
3.10	$z = 8$ の時の出力結果 . . . . .	25
3.11	$j = 2.0$ 、 $k = 1.0$ の時の出力結果 . . . . .	26
3.12	$j = 20.0$ 、 $k = 19.0$ の時の出力結果 . . . . .	26

3.13	$j = 20.0$ 、 $k = 1.0$ の時の出力結果 . . . . .	26
3.14	$d = 5.0$ の時の出力結果 . . . . .	27
3.15	$d = 20.0$ の時の出力結果 . . . . .	27
3.16	出力結果 . . . . .	28
3.17	出力結果のズーム . . . . .	29
3.18	実際の古紙 . . . . .	29
3.19	折り変形を加えた出力モデル . . . . .	30
3.20	折り変形箇所のズーム 視点 1 . . . . .	30
3.21	折り変形箇所のズーム 視点 2 . . . . .	30
3.22	折り変形箇所のズーム 視点 3 . . . . .	30
3.23	内部に頂点がない出力結果 . . . . .	32

# 第 1 章

## はじめに

### 1.1 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」）技術を用いたアニメ、映画、ゲームなどの作品（以下「コンテンツ」）の開発が盛んに行なわれている。3DCG コンテンツでは登場する人物や物を3次元形状モデル（以下「モデル」）で表現している。モデルを制作するためには、物体の形状、位置、大きさなどをコンピュータ内部で表現する必要があり、この作業をモデリングと呼ぶ[1]。モデリングを行うためのソフトをモデリングツールと呼び、代表的なモデリングツールとしてはMetasequoia[2]、Maya[3]、Blender[4]などがある。モデリングを行う際は、いかに現実の物らしく表現するかが重要な点の1つである。加えて、モデルに色や模様を描く画像（以下「テクスチャ」）を貼り付けることによって3DCG コンテンツに登場するものは制作する。対象物のリアリティを表現するためには、物体が持つ外見上の性質をモデルとテクスチャで再現することにより表現する。モデリングをする上で、現実の劣化の性質を再現することで、そのモデルが古い物であることを示すことがある。

物によって劣化の仕方は異なり、劣化処理の研究としては建造物の酸性雨劣化[5]や、石の風化の研究[6]や、金属のさびの研究[7]があるが、本研究は紙に対する劣化を対象とする。紙は情報の伝達手段や、包装など日々の生活と密な関係にある[8]。そのため、コンテンツ上でも登場する機会は多い[9][10]。中でも紙を長

い間使用した、あるいは保管したなど時間が立つことによる劣化である経年劣化 [11][12][13][14][15] を起こした紙は、コンテンツの物語上で重要なアイテムとして扱うことも多い。本来、古紙とは一度使用した紙でリサイクル可能な紙をさすが、本論では経年劣化を起こし切れ目や破れ目など形状が変化した紙のことを古紙と定義する。図 1.1 にコンテンツ用に制作された古紙テクスチャの一例を示す [16]。

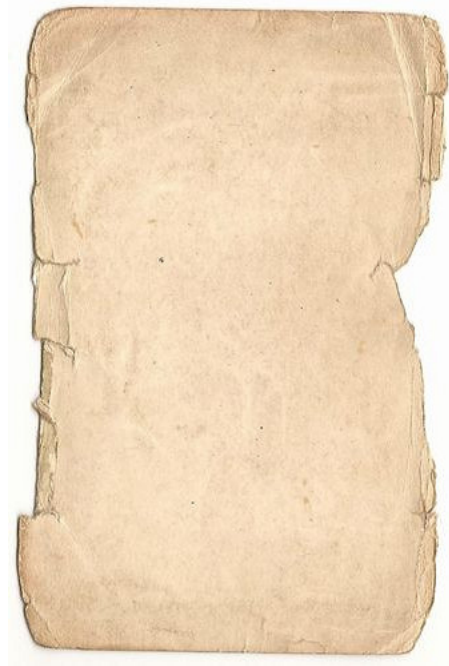


図 1.1: コンテンツ用に制作された古紙テクスチャの一例

本研究では古紙の外見に表れる劣化として切れ目、破れ目、折れ目、湾曲、変色の要素に着目した。図 1.2 は劣化の要素が確認できる実際の古紙の一例を示す [17]。

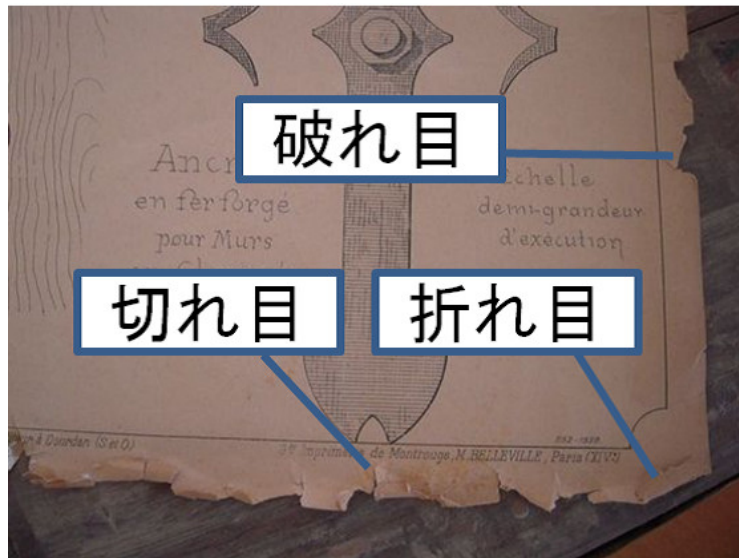


図 1.2: 劣化が確認できる古紙の一例

本研究の対象とする切れ目、破れ目について述べる。切れ目は、外的な力により紙の繊維を引き伸ばすことで、絡み合った繊維同士がほどけてしまい、紙に断裂が入った状態になる劣化である。破れは、切れによってできた断裂が、他の断裂と交差することで紙の一部が欠落した状態になる劣化である。

劣化表現の制作において、3DCG コンテンツにおいて古紙を制作する際に一般的にどのような方法で古紙モデルを制作しているかを述べる。紙の形状をモデルで表現する際は、モデルを構成する多角形（以下「ポリゴン」）に対して、劣化した状態の紙を描いた画像テクスチャ（以下「古紙テクスチャ」）のマッピングを行い劣化表現を行うのが一般的である。例えば、切れや破れの劣化表現を行いたければ、古紙テクスチャの切れ目や破れ目として欠落させたい部分を透明化することで表現する。この手法の利点として、細かい切れ目や、破れ目などの表現をモデリングでの制作に比べて非常に簡単に行えることが利点である。また、モデリングに比べてテクスチャへ劣化表現を描くのは絵を描くことと同じ要領で編集することが可能であり、比較的容易であることも利点である。

一方で、折れ目の劣化表現をしたい場合など、様々な角度から古紙テクスチャ



を見る時には、古紙テクスチャを貼り付ける前にポリゴンを変形させる工程が必要になる。同じくポリゴンの変形が必要となる表現として、紙の湾曲や、紙がばらばらになる劣化がある。こうしたポリゴンの変形において、切れ目や破れ目がある箇所から折れ目ができる状態を立体的に表現しようとする、テクスチャ上ではなくポリゴン上で切れ目、破れ目を制作しなければならない。しかし、ポリゴンの変形で切れ目、破れ目の劣化表現を制作する工程は手間と時間がかかる。

そこで、本研究では折れ目や湾曲の劣化表現が古紙テクスチャによる表現では意図しない結果になりやすく、モデリングによる表現では手間や時間がかかってしまうことを問題とした。この問題を解決し、3DCG コンテンツを制作する上で古紙のモデリングを効率化することを目的とする。

既存研究として、3DCG モデルにおいて紙の劣化表現を行なっている様々な手法がある。佐藤ら [18] は紙のリアルな破断表現を目的として、ポリゴンが全て三角形で表現したモデルに対して有限要素法を適用することで破断位置と方向を計算し、生成した亀裂に応じてメッシュを再構築することで破断をシミュレーションしている。有限要素法とは、対象を有限の要素で分割し全体を要素の集合体として表現する手法である。結果として、シミュレートした結果にノイズ関数を用いたテクスチャを貼り付けることで破れ目のゆらぎを表現した。ノイズ関数は、パラメータとして整数を取り、そのパラメータに基づいた乱数を返す関数のことである。計算処理にかかる時間も精度を要求しなければ1ステップあたり0.7秒というほぼリアルタイムに近い計算速度を実現している。

木谷ら [19] も同じくリアルな紙の切断表現を目指し、紙の繊維をブロック構造として多層化したモデルを作成し、立方体格子の各頂点に質点、各辺と各対角線にばねを配置した質点-ばね格子で表現した弾性体モデルで紙を表現した。これにより切断時の表層だけが剥がれる様子や、破れ目のゆらぎを表現した。

大森 [20] は、この木谷らの研究で多層構造のため発生していた紙が伸びたり、厚くなったりする問題を単層構造で作成することで解決した。表層だけ剥がれる様子の表現は、紙の繊維分布画像を多数用意し、それら画像をずらして貼り付ける

ことで実現した。

前述した佐藤ら、木谷ら、大森の研究は紙の切れ目をアニメーションで表現することを目的の一つとしており、細かい切れ目、破れ目を多数生成することを目的としていない。対して、本研究目的ではモデリングの制作支援であり、紙が切れていく様子をアニメーションで表現する必要はない。そのため、より簡易な手法で紙の切れ目が表現できる。

岡ら [21] は、波打ち状に湾曲する劣化に対して紙モデルを分割することで制御点を求め、それを基にベジエ曲面で湾曲を表現し自動処理で形状変形を行うツールを開発している。ベジエ曲面の説明の前にベジエ曲線について述べる。ベジエ曲線とは、4点の制御点と制御点を結んで得る3つの線分の内分点を求め、求めた2点の内分点を結んで得る線分の内分点を繰り返し求めることで得る曲線のことである。ベジエ曲面はベジエ曲線において1次元であった内分点の値を2次元に拡張して求めるものとなる [22]。

木曾ら [23] は Perlin Noise を用いることで擬似的なしわの表現を実現した。Perlin Noise は多様なノイズ関数を平滑化し、加算したものである。また、観測により紙のしわを定量的に測定しその値を基にした更にリアルなしわ表現の可能性を述べている。

前述した岡ら、木曾らの研究では、湾曲の劣化を自動処理で表現することができているが、切れ目や、破れ目といった欠損の表現は確認できない。そこで、本研究目的に対して、モデリングにおいて手間や時間のかかる切れ目と破れ目の劣化の制作工程を自動で処理する手法を提案する。また、提案した手法を用いて古紙の劣化表現制作支援ツールを作成する。制作支援を目的とするため、本研究で開発ツールでは汎用的なモデリングツールで編集可能なデータ形式で出力することとする。汎用的なモデリングツールとして Metasequoia を選択し、出力するファイル形式は mqo ファイルとする。劣化処理中は、ハーフエッジデータ構造をとり、出力時にインデックスフェースセットへと変換する。ハーフエッジデータ構造は、「頂点、稜線、半稜線、ループの四種類の位相要素の集合によって表現するソリッ

ドモデル用データ構造。」(渡辺, 2002, p.1) と定義される。インデックスフェースセットは「配列による頂点座標データの集合と、頂点番号による多角形データの集合に二つからなるソリッドモデル用データ構造。」(渡辺, 2002, p.1) と定義される [24]。

提案手法は、紙の周囲部分でのみ頂点の生成、移動、削除を行うことで変形にかかる計算量を減らし、劣化表現生成までの時間を短縮するものである。また、手法を実装し動作検証することで手法の有効性と展望を示す。

## 1.2 論文構成

本論文の構成は次の通りである。2章では紙の切れ目、破れ目、破れ目からの切れ目を生成する手法について述べる。3章では今回提案した手法の検証、考察について述べる。

## 第 2 章

# 切れ目、破れ目の生成の提案手法

本章では、本研究における切れ目、破れ目を生成する手法について述べる。

### 2.1 劣化表現生成手法の概要

本手法は紙モデルを構成するポリゴンを変形させることによって切れ目、破れ目、破れ目からの切れ目、の劣化を表現することを目的とする。切れ目を頂点移動によって生成し、破れ目を頂点削除によって生成し、破れ目からの切れ目を生成する準備や、多様な切れ目の生成の際に頂点生成を用いる。処理にあたっては、二次元平面上で処理を行い、出力時に三次元データとして扱えるようにする。

### 2.2 劣化表現生成手法の流れ

まず劣化していない状態の紙を表す図形を生成する。この図形を基本図形と定義する。この基本図形に対して、ポリゴンの変形を行うことで劣化した紙モデルを表現する。

次に基本図形に対して行う劣化表現について述べる。切れ目の劣化処理、破れ目の劣化処理、破れ目から伸びる切れ目の劣化処理が存在し、これら3種類の処理の全てを一連の劣化処理とする。また、一連の劣化処理において劣化処理を行う順番を述べる。破れ目は切れ目によって繊維が完全に断裂して欠落する劣化の

ため、切れ目の生成後に生成する。破れ目から伸びる新たな切れ目は破れ目がないと生成できないため、破れ目の生成後に生成する。よって、劣化の生成する順番は切れ目、破れ目、破れ目から伸びる切れ目の順である。劣化処理の一連の流れを繰り返し行うことで、多様な劣化表現が可能である。

従って全体的な流れとしては、最初に基本図形の生成を行い、基本図形に対してステップを重ねていくということになる。図 2.1 は本手法の全体的なフローチャートである。

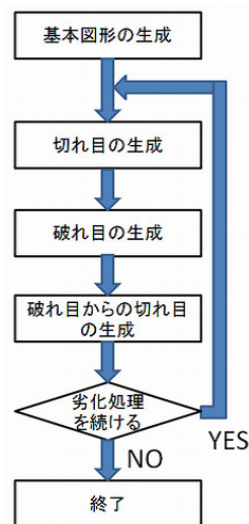


図 2.1: 手法のフローチャート

## 2.3 手法で用いるパラメータについて

本手法ではユーザーが操作するパラメータを用意する。このパラメータによって様々な結果が出力の操作を可能にする。次に操作するパラメータについて述べていく。第1のものに、劣化の発生しやすさ  $p$  がある。劣化の処理が発生するかどうかを頂点ごとに判定するが、 $p$  はその発生確率である。 $p$  は 0~1 の範囲内の値をとる。第2のものに、切れ目の幅の乱数の最大値  $j$  と第3のものに、切れ目の幅の乱数の最小値  $k$  がある。切れ目の劣化処理では切れ目の幅を乱数で求める。その

切れ目の幅を決定する乱数の最大値と最小値を決定するパラメータである。 $j$ は2～20の範囲内の値をとり、 $k$ は1～19の範囲内の値をとる。第4のものに切れ目の角度のノイズを $z$ がある。切れ目を入れる際に、その切れ目にノイズを足すことで多様な角度を持った切れ目を生成する。そのノイズの値が $z$ である。 $z$ は1～8の範囲内の値をとる。第5のものに、切れ目の長さの乱数の最大値 $m$ と第6のものに、切れ目の長さの乱数の最小値 $n$ がある。切れ目の劣化処理では切れ目の長さを乱数で求める。その切れ目の長さを決定する乱数の最大値と最小値を決定するパラメータである。 $m$ は2～20の範囲内の値をとり、 $n$ は1～19の範囲内の値をとる。第7のものに、破れ目のなりやすさを $d$ がある。破れ目は切れ目同士をつなぐことで生成する。切れ目同士をつなぐかどうかは距離によって判定するが、この距離の値が $d$ である。 $d$ は0～20の範囲内の値をとる。これらパラメータの範囲は、意図しない出力結果を避けるために暫定的に定めたものである。表2.1にパラメータの一覧を示す。

表 2.1: ユーザが操作するパラメータ一覧

パラメータ名	範囲	用途
$p$	0.0～1.0	劣化の発生確率
$j$	2.0～20.0	切れ目の最大幅
$k$	1.0～19.0	切れ目の最小幅
$z$	1.0～8.0	切れ目の角度のノイズ
$m$	2.0～20.0	切れ目の最大長
$n$	1.0～19.0	切れ目の最小長
$d$	0.0～20.0	切れ目の接続距離

## 2.4 基本図形の生成

まず、劣化する前の紙として正方形の図形を生成する。正方形は $xy$ 平面上に4点の頂点で構成する。この4点を左上の頂点から反時計回りの順にそれぞれ頂点 $A$ 、頂点 $B$ 、頂点 $C$ 、頂点 $D$ と定義する。各頂点は $A$ は $(-1, 1)$ 方向、 $B$ は $(-1, -1)$

方向、 $C$  は  $(1, -1)$  方向、 $D$  は  $(1, 1)$  方向にあるものとする。4 点で構成した正方形を示す図が図 2.2 である。

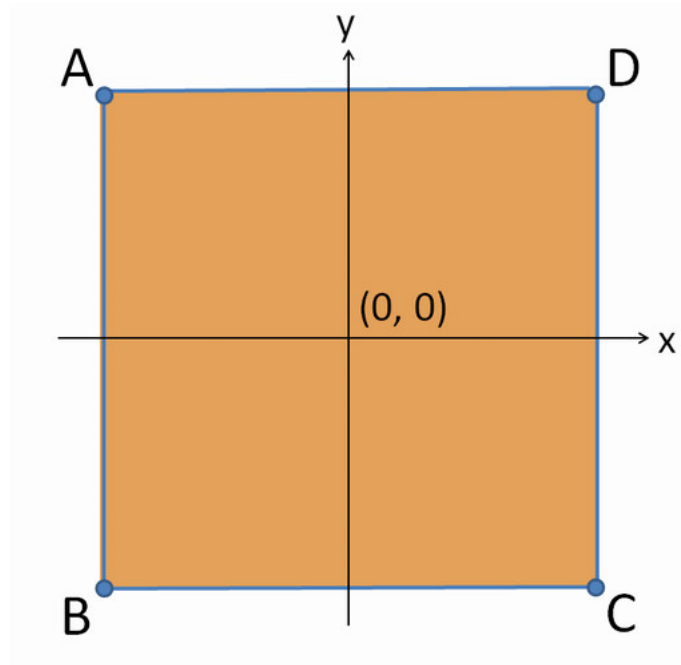


図 2.2: 4 点で構成した正方形

次に、各辺に対して新たな頂点を生成する。新たな頂点の生成が終了した時点で基本図形を完成とし、次の工程へ進む。図 2.3 は全頂点数が 128 の時の図形を示している。

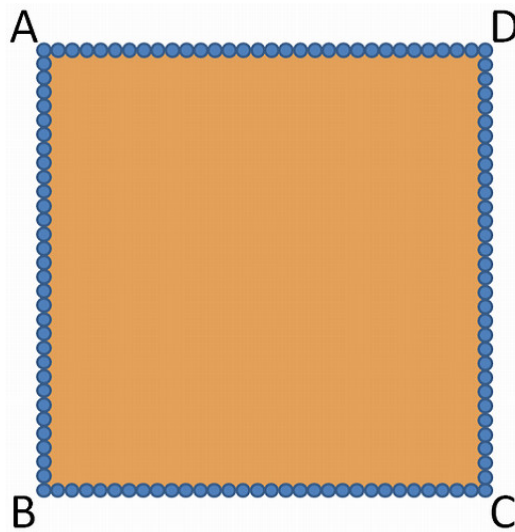


図 2.3: 頂点を新たに生成した図形

## 2.5 切れ目の生成

切れ目の劣化を生成するために、まずどの部分から切れ目が入るかを決定する。そのために、切れ目の開始地点の頂点を抽出する。抽出した頂点がモデルの内側へ移動することで大まかな切れ目を表現できるが、切れ目の形状としては、欠損している面積が広すぎてしまい切れ目の表現には見えない。そこで、抽出した頂点位置に近い位置に2点の新たな頂点を生成することで切れ目の幅となる部分を生成する。その上で、抽出した頂点を移動させることで切れ目の表現を生成する。2.5.1節より、各工程について詳細に述べていく。

### 2.5.1 切れ目の開始地点の決定

基本図形の各頂点から切れ目となる頂点を求める。全頂点の内、ランダムで抽出した頂点を切れ目となる頂点とし、抽出した頂点を切れ目頂点と定義する。頂点を切れ目頂点として抽出する確率はユーザが制御する。ユーザが制御する0~1の範囲の値を $p$ とした時、 $p$ の確率で頂点を切れ目頂点とする。



## 2.5.2 切れ目の幅の生成

図2.4は切れ目頂点に対して、接続している2辺の midpoint となる位置に新たに頂点を生成したことを示している。

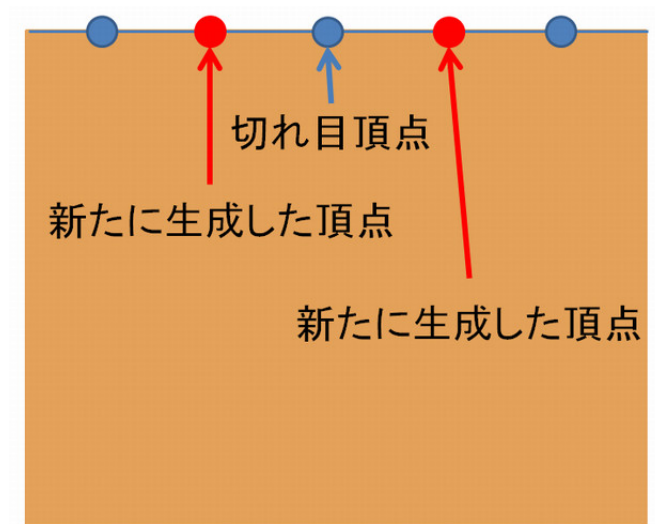


図2.4: 切れ目の幅を作るために新たに頂点を生成

新たに生成した頂点を、切れ目頂点に対して一定の距離まで近づける。近づける方法として、新たに生成した頂点と切れ目頂点の内分点を利用する。内分点で利用する比率は  $1:k$  とし、 $k$  の値が大きければ大きいほど切れ目頂点に近づくことを示す。 $k$  は  $m$  以上  $n$  以下の任意の整数をランダムに抽出して設定する。 $m$  と  $n$  の値はユーザの任意の値をとる。また、内分点の比率は切れ目1つごとに再計算する。よって、新たに生成した2点の頂点の移動に利用する内分点の比率は同じである。図2.5は求めた内分点の座標に新たに生成した2点の頂点を移動したことを示すものであり、これによって切れ目の幅が算出する。

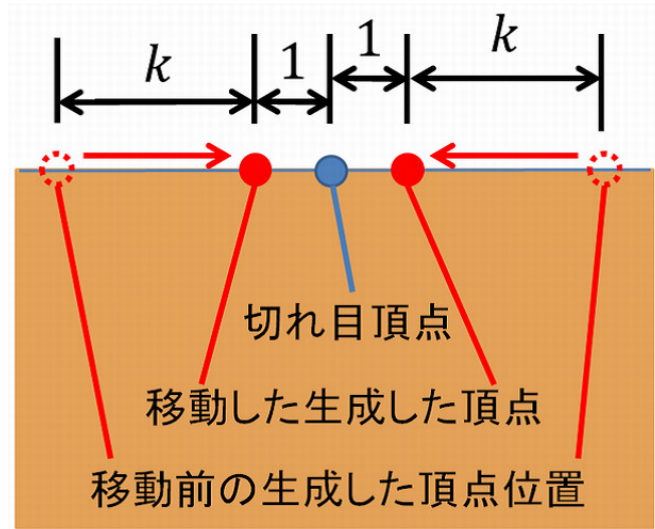


図 2.5: 切れ目の幅の決定

### 2.5.3 切れ目の角度の決定

切れ目頂点を 2.4 節で定義した各頂点に対して切れ目を入れる角度を設定する。切れ目は各辺の位置と対となる方向へ伸ばしていく。例えば、上に位置する辺  $AD$  の対となる辺は下の辺  $BC$  であり、辺  $AD$  に属する頂点は切れ目は辺  $BC$  に向かって伸びていく。こうした方向を各頂点に設定するため、原点から見て辺  $CD$  方向を  $0^\circ$  とし、各頂点へ切れ目が伸びるべき方向を次のように定義する。すなわち、頂点  $A$  は  $315^\circ$ 、頂点  $B$  は  $45^\circ$ 、頂点  $C$  は  $135^\circ$ 、頂点  $D$  は  $225^\circ$ 、辺  $AD$  に属する頂点群は  $270^\circ$ 、辺  $AB$  に属する頂点群は  $0^\circ$ 、辺  $BC$  に属する頂点群は  $90^\circ$ 、辺  $CD$  に属する頂点群は  $180^\circ$  である。

その角度に微小な数値のノイズを付加することで、様々な角度に切れ目が入るようにする。ノイズはユーザが制御する任意の正の値  $z$  を最大値とし、 $-z$  を最小値とした乱数から算出した値を角度に加算することで付加する。

## 2.5.4 切れ目頂点の移動量の決定と切れ目の生成

角度を決定した後、その角度の方向に切れ目の長さ分だけ頂点を進めることで切れ目が生成する。切れ目の長さは  $j$  以上  $k$  以下の任意の生の値をランダムに抽出して設定する。 $j$  と  $k$  の値はユーザの任意の値をとる。

求めた角度と切れ目の長さの2値を利用して実際に頂点を移動させる。移動前の頂点座標位置ベクトルを  $\mathbf{E}$ 、 $\mathbf{E}$  の  $x$  成分を  $q$ 、 $\mathbf{E}$  の  $y$  成分を  $r$ 、移動後の頂点座標位置ベクトルを  $\mathbf{F}$ 、 $\mathbf{F}$  の  $x$  成分を  $s$ 、 $\mathbf{F}$  の  $y$  成分を  $t$ 、角度を  $\theta$ 、切れ目の長さを  $l$  とした時、式 (2.1) で求める。

$$\begin{pmatrix} s \\ t \end{pmatrix} = \begin{pmatrix} q \\ r \end{pmatrix} + l \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix} \quad (2.1)$$

この時、 $l(\cos \theta, \sin \theta)$  を切れ目の移動量の情報として記憶しておき、この情報を  $V$  と定義する。この移動を行った様子を示しているのが図 2.6 である。

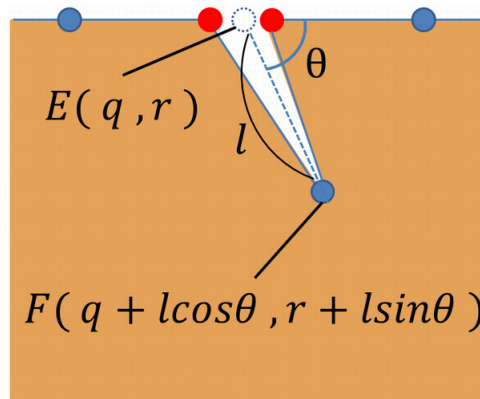


図 2.6: 頂点の移動による切れ目の生成

## 2.5.5 切れ目の伸長

切れ目の生成は 2.5.1 節から 2.5.4 節までの通りだが、単なる生成処理だけでは直線的な切れ目しか生成できない。そこで生成した切れ目を角度をつけて伸ばし

ていくことで様々な切れ目を生成する。切れ目の伸長方法として、2通りの方法がある。第1のものは、2.5.2節で用いた内分比を利用する方法である。これを内分比手法と定義する。第2のものは、予め記憶しておいた切れ目頂点の移動量を利用する方法である。これを移動量手法と定義する。本提案手法では移動量手法を選択した。理由として、内分比手法の場合延長した切れ目の幅が延長する程に狭くなっていき、1回目の延長の時点でも出力した時に切れ目のように見えない程になってしまう問題があるためである。移動量手法の場合は切れ目の幅は切れ目の生成の時の幅が維持されるため、この問題は発生しない。それぞれの手法で劣化表現をした際の様子を図2.7と図2.8に示す。

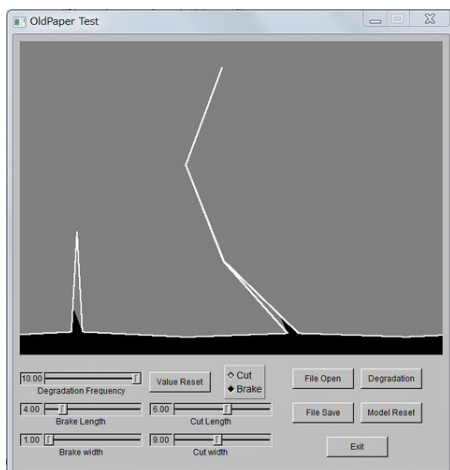


図 2.7: 内分比手法による切れ目

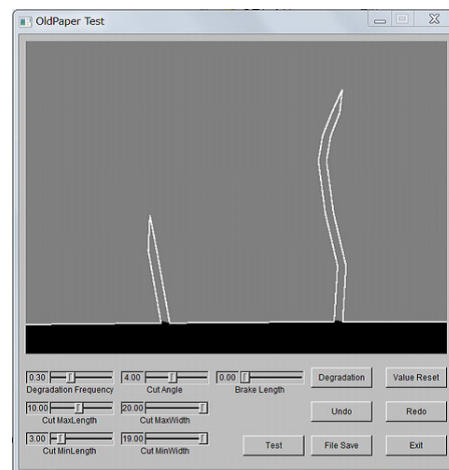


図 2.8: 移動量手法による切れ目

次より移動量手法について述べる。まず、切れ目頂点に辺を挟んで隣接している2つの頂点を頂点 **Q**、**R** と定義する。次に、切れ目頂点と頂点 **Q**、**R** それぞれの中点を取り、その中点座標に新たに頂点を生成する。

図2.9は切れ目の延長において節目となる新たな頂点を2つ生成したことを示す。

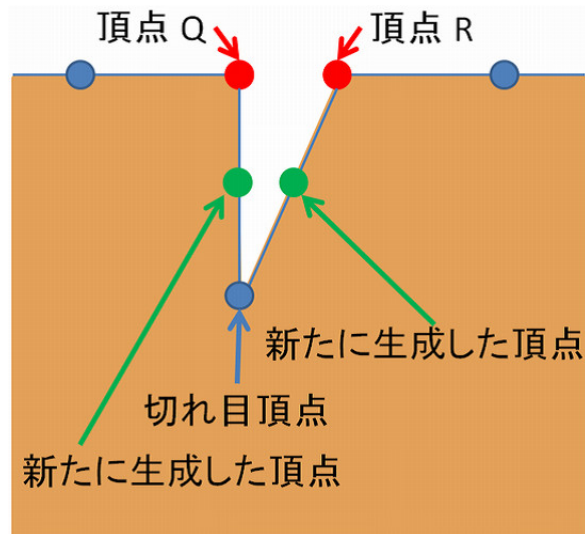


図 2.9: 切れ目の伸長用の頂点を生成

これらの新たな頂点を頂点  $Q$ 、 $R$  の位置にそれぞれ移動した後、2.5.1 節で取得しておいた切れ目頂点の移動量に従って更に移動させる。新たな頂点の移動先を  $S$ 、 $T$ 、2.5.4 節で記憶した切れ目頂点の移動量を  $V$  とした時、これら移動先の位置座標を次の式 (2.2) で求める。

$$S = Q + V, \quad T = R + V \quad (2.2)$$

これによって、切れ目頂点の位置の近くに新たな頂点 2 つを生成しているため、切れ目頂点を移動することによって切れ目の延長を行う。

切れ目頂点の移動処理については 2.5.1 節同様に行う。切れ目頂点が移動することによって切れ目が延長できた様子を図 2.10 に示す。

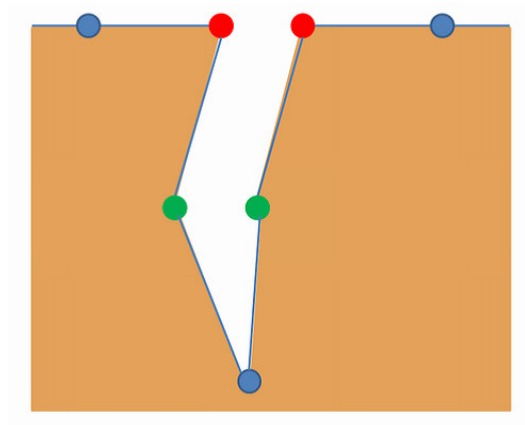


図 2.10: 切れ目の伸長

## 2.6 破れ目の生成

破れ目は切れ目の先端に位置する頂点同士の距離  $e$  が、0 以上の任意の値  $d$  以下になった時に、頂点を辺で結び、切れ目同士をつながってできた新たな面を削除することによって生成する。この時生成した新たな辺には「破れ目の辺である」という情報を保存しておく。頂点同士を辺でつなぐことによって新たな面を生成するが、この時削除する面を判定する方法として、面がもつ頂点数を比較し、少ない方を削除している。図 2.11 は、切れ目頂点同士を辺で結んでいること示し、図 2.12 は、結んだ辺によって生成した破れ目を示している。

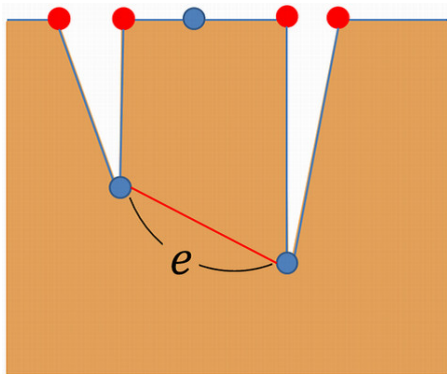


図 2.11:  $e$  の値が  $d$  の値以下の時に  
頂点を辺で結ぶ

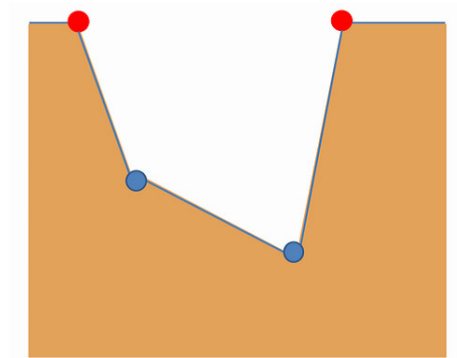


図 2.12: 辺を結ぶことで新たにできた面  
を削除し破れ目を生成

## 2.7 破れ目からの切れ目の生成

2.6 節において保存しておいた「破れ目の辺である」という情報を持つ辺を抽出する。抽出した辺に対して、2.4 節と同様に辺を等分して頂点を生成していく方法により、破れ目からの切れ目の開始点となる頂点を増やしていく。これら破れ目の辺に新たに生成した頂点は切れ目になることが可能であるという情報を記憶しておく。それら頂点から切れ目とする頂点を抽出し、生成に至るまでを 2.5.1 から 2.5.4 節まで同様に行う。

図 2.13 は、破れ目の辺に対して頂点を新たに生成している様子を示し、図 2.14 は、破れ目から切れ目を生成した様子を示している。

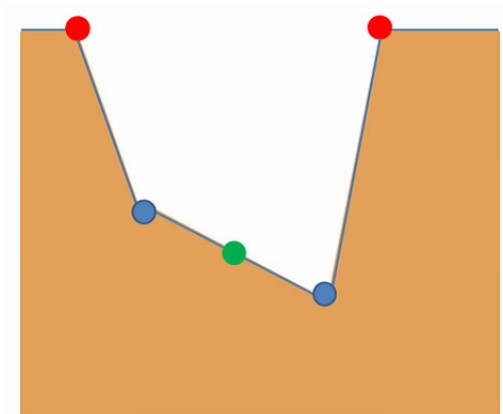


図 2.13: 破れ目に新たな頂点を生成

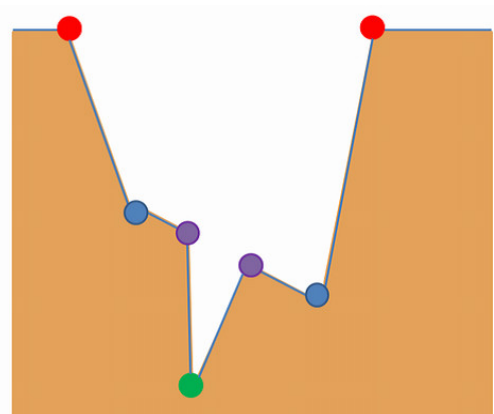


図 2.14: 生成した頂点から切れ目を生成



# 第 3 章

## 動作検証

### 3.1 実装

2 章で述べた本研究の手法を用いて実際にツールを開発した。今回の開発では、開発言語として C++ を使用し、実装には 3 次元グラフィックツールキットである「Fine Kernel Tool Kit」 [25] を利用した。

### 3.2 ツール概要

図 3.1 がツール起動時の画面である。

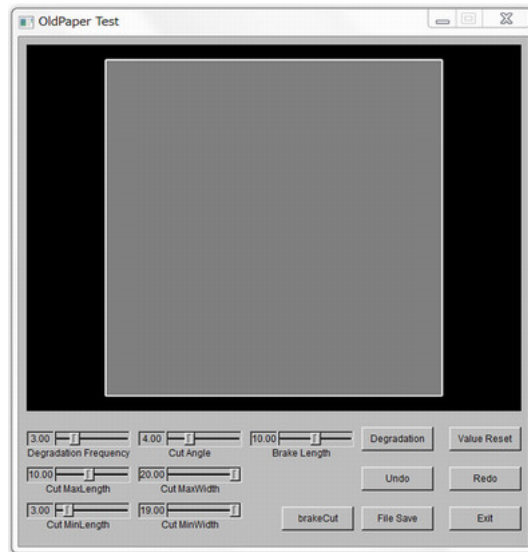


図 3.1: 紙の劣化自動生成ツール

画面下部のスライダーとボタンを操作することで様々な劣化変形が可能である。劣化の起こりやすさ、切れ目の長さの乱数の最小値と最大値、切れ目の幅の乱数の最小値と最大値、切れ目の角度のブレやすさ、破れ目になりやすさといったパラメータをスライダーで操作する。スライダーで決定したパラメータを基に Degradation ボタンを押すことで劣化処理を実行する。この操作を複数回繰り返すことで、満足する出力結果が出たら File Save ボタンで mqo ファイルを出力し、Exit ボタンでツールを終了する、という流れになる。評価にあたって、2.4 節で述べた処理である頂点の生成数は多ければ多いほど、細かい劣化の処理が可能となるがその反面、処理速度が落ちてしまう。一方で、頂点数が少なすぎると劣化処理を行える場所が少なくなってしまう。これらを考慮した上で、各辺を 32 等分した。

2.3 節と同じく操作するパラメータとして劣化の起こりやすさの確率を  $p$ 、切れ目の幅の乱数の最大値と最小値を  $j$  と  $k$ 、切れ目の角度のブレやすさを  $z$ 、切れ目の長さの乱数の最大値と最小値を  $m$  と  $n$ 、破れ目になりやすさを  $d$  とする。

### 3.3 評価

まず、2.3節で述べた各パラメータによってどのように出力結果に差が生じるかを述べる。劣化の起こりやすさである  $p$  を操作した時の出力結果を次に示す。 $p$  は2.5.1節の処理で利用している。図3.2、図3.3、図3.4はそれぞれ  $p$  が0.1、0.3、1.0の時の出力結果を示している。なお、他のパラメータは3つの出力結果それぞれにおいて同値である。

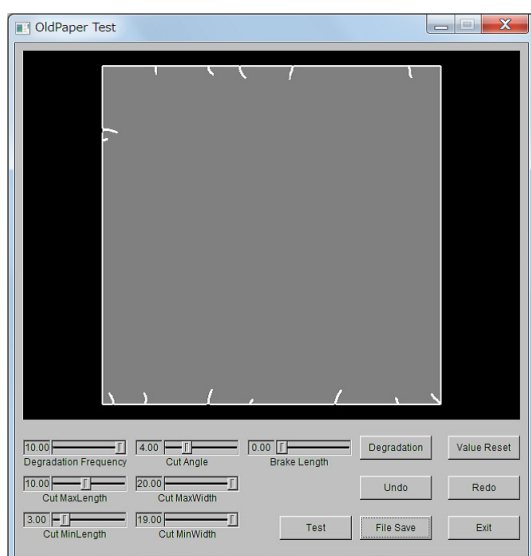


図 3.2:  $p = 0.1$  の時の出力結果

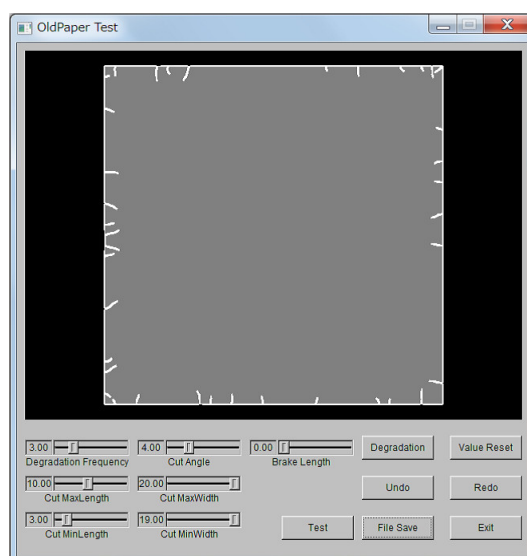


図 3.3:  $p = 0.3$  の時の出力結果

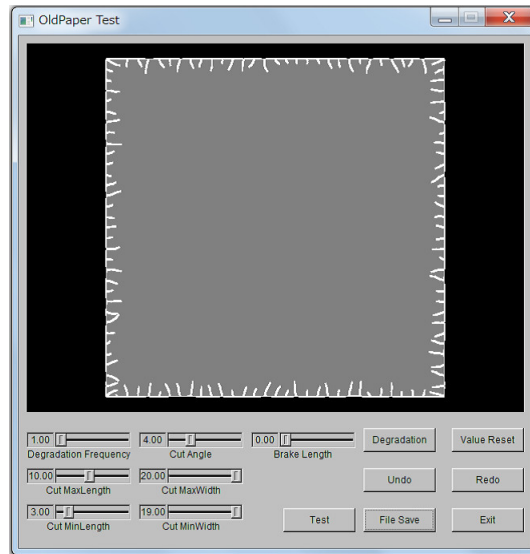


図 3.4:  $p = 1.0$  の時の出力結果

切れ目の幅に影響する  $m$  と  $n$  を操作した時の出力結果を次に示す。 $m$  と  $n$  は 2.5.2 節の処理で利用している。図 3.5、図 3.6、図 3.7 はそれぞれ  $m = 2$  かつ  $n = 1$ 、 $m = 20$  かつ  $n = 19$ 、 $m = 20$  かつ  $n = 1$  の時の出力結果を示している。なお、他のパラメータは 3 つの出力結果それぞれにおいて同値である。

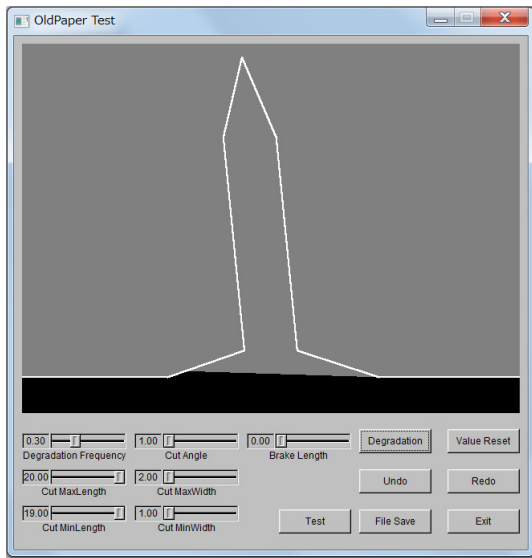


図 3.5:  $m = 2.0$ 、 $n = 1.0$  の時の出力結果

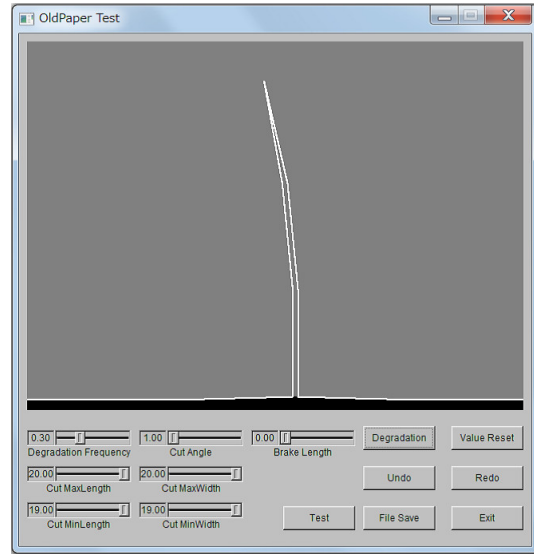


図 3.6:  $m = 20.0$ 、 $n = 19.0$  の時の出力結果

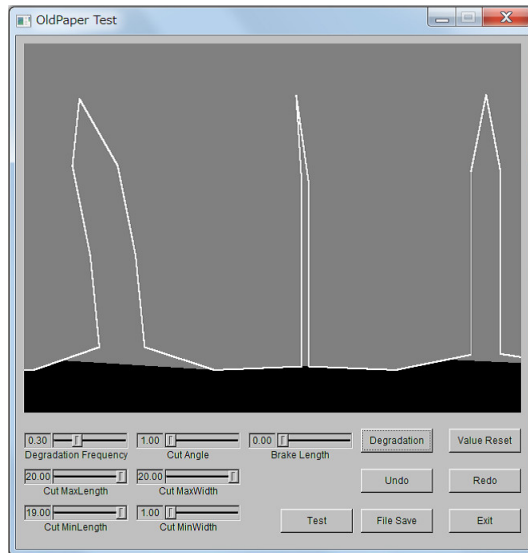


図 3.7:  $m = 20.0$ 、 $n = 1.0$  の時の出力結果

切れ目の角度のノイズである  $z$  を操作した時の出力結果を次に示す。 $z$  は 2.5.3 節の処理で利用している。図 3.8、図 3.9、図 3.10 はそれぞれ  $z = 1.0$ 、 $z = 4.0$ 、

$z = 8.0$  の時の出力結果を示している。なお、他のパラメータは3つの出力結果それぞれにおいて同値である。

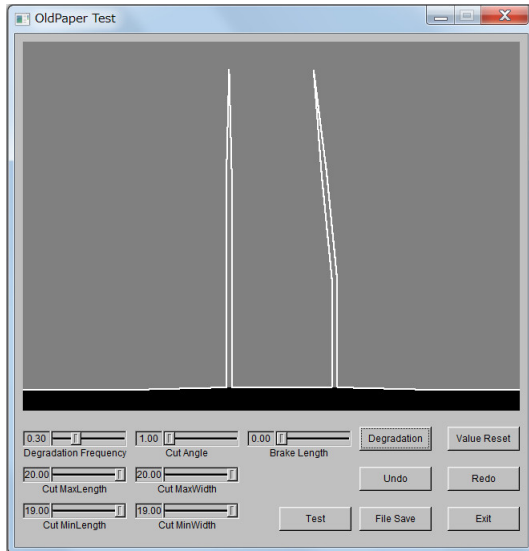


図 3.8:  $z = 1.0$  の時の出力結果

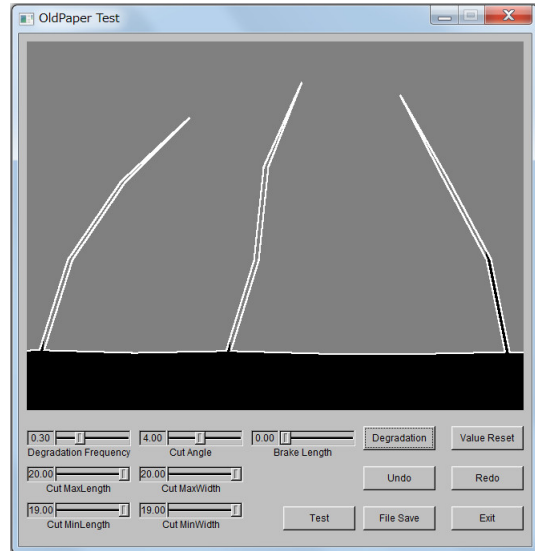


図 3.9:  $z = 4.0$  の時の出力結果

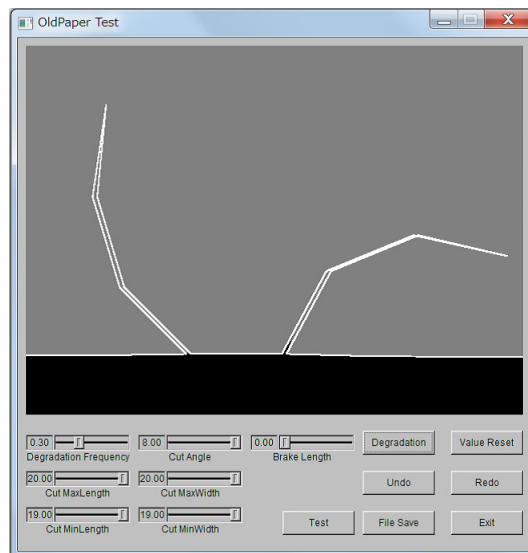


図 3.10:  $z = 8$  の時の出力結果

切れ目の長さに影響する  $j$  と  $k$  を操作した時の出力結果を次に示す。 $j$  と  $k$  は

2.5.4 節の処理で利用している。図 3.11、図 3.12、図 3.13 はそれぞれ  $j = 2.0$  かつ  $k = 1.0$ 、 $j = 20.0$  かつ  $k = 19.0$ 、 $j = 20.0$  かつ  $k = 1.0$  の時の出力結果を示している。なお、他のパラメータは3つの出力結果それぞれにおいて同値である。

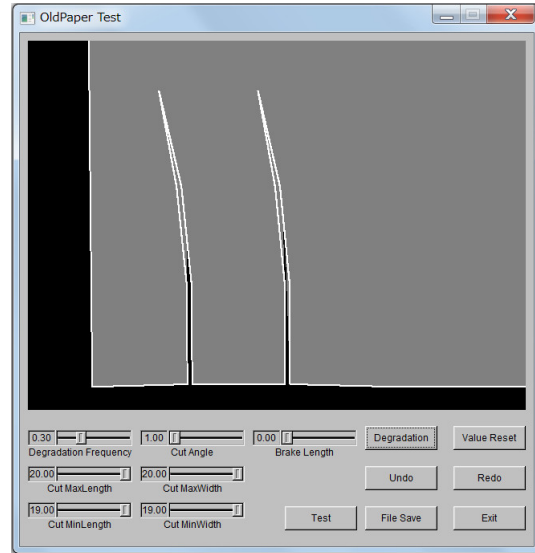
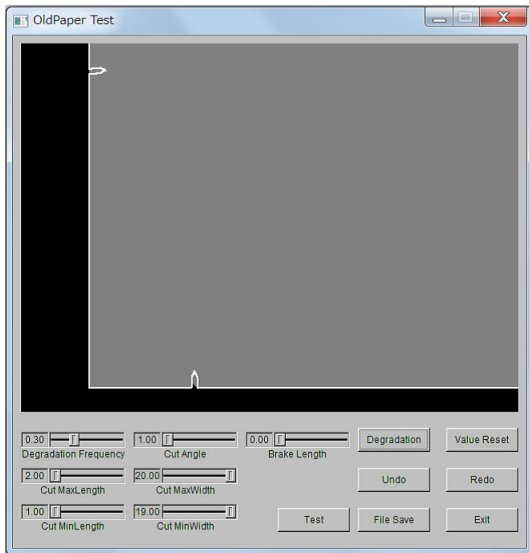


図 3.11:  $j = 2.0$ 、 $k = 1.0$  の時の出力結果      図 3.12:  $j = 20.0$ 、 $k = 19.0$  の時の出力結果

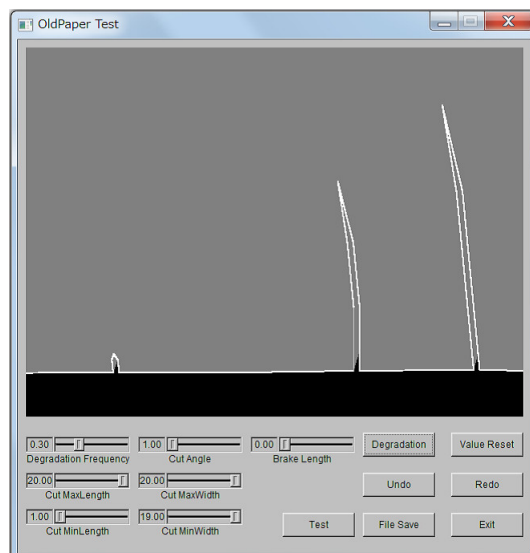


図 3.13:  $j = 20.0$ 、 $k = 1.0$  の時の出力結果

破れ目のなりやすさである  $d$  を操作した時の出力結果を次に示す。 $d$  は 2.6 節の処理で利用している。図 3.14、図 3.15 はそれぞれ  $d = 5.0$ 、 $d = 20.0$  の時の出力結果を示している。なお、他のパラメータは 2 つの出力結果それぞれにおいて同値である。

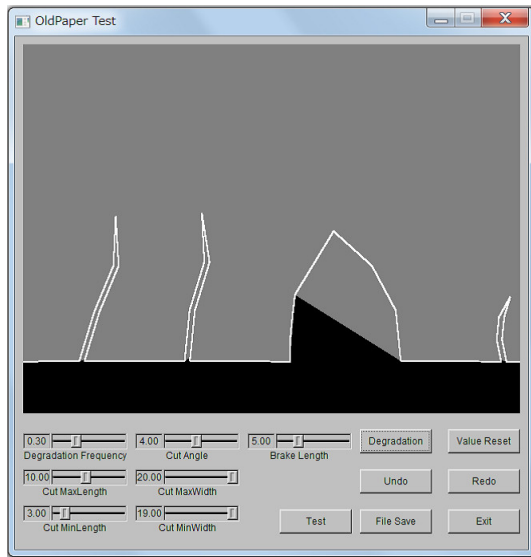


図 3.14:  $d = 5.0$  の時の出力結果

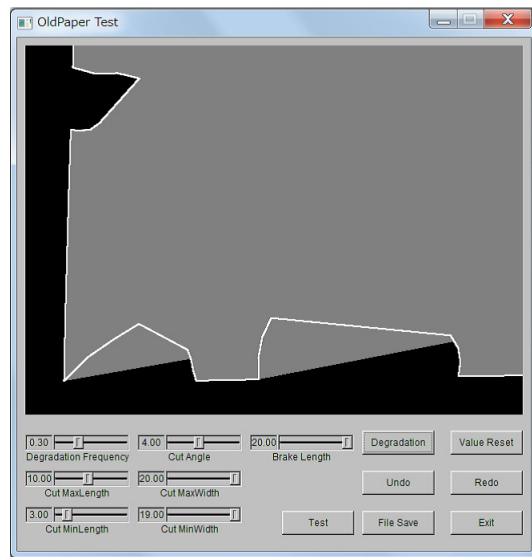


図 3.15:  $d = 20.0$  の時の出力結果

次に mgo ファイルとして出力について述べる。図 3.16 はツールによって出力した mgo ファイルを Metasequoia によって読み込んだ画面である。出力の際のパラメータを表 3.1 に示す。





図 3.16: 出力結果

表 3.1: 図 3.16 出力時のパラメータ

劣化の起こりやすさ $p$	0.6
切れ目の幅の細さの度合いの最大値 $m$	20.0
切れ目の幅の細さの度合いの最小値 $n$	19.0
切れ目の角度のノイズ $z$	4.0
切れ目の長さの最大値 $j$	10.0
切れ目の長さの最小値 $k$	3.0
破れ目のために切れ目同士をつなぐ長さ $d$	10.0

更に図 3.17 は出力したモデルの一部を拡大表示したものである。図 3.18 は実際の古紙を写した画像であるが、両者の画像を比較すると切れ目の劣化、破れ目の劣化、破れ目から切れ目が生じる劣化をそれぞれ再現できていることがわかる。



図 3.17: 出力結果のズーム

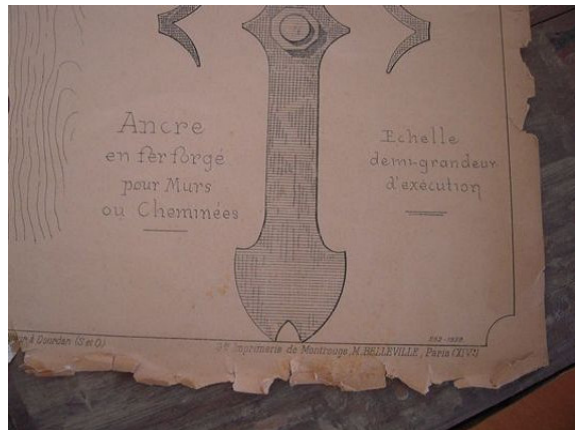


図 3.18: 実際の古紙

次に、出力したモデルがポリゴンの変形によって作ることで折れの劣化表現を加えることが可能なことを示す。折れの劣化表現を加えるのは、制作支援対象として挙げた Metasequoia を利用する。Metasequoia に備わっている曲げ機能を利用することにより、ポリゴンを曲げる要領で形状変形を行うことが可能である。図 3.19 は出力したモデルを曲げ機能を利用して変形した様子を示している。図 3.20、図 3.21、図 3.22 は図 3.19 の赤い丸で囲んだ箇所をズームし、3つの視点から見たものである。ポリゴン自体に変形を加えたため、テクスチャで劣化を表現した場合と異なり立体的に折れ目劣化が表現できていることがわかる。



図 3.19: 折り変形を加えた出力モデル



図 3.20: 折り変形箇所のズーム 視点 1



図 3.21: 折り変形箇所のズーム 視点 2



図 3.22: 折り変形箇所のズーム 視点 3

本手法を実装したプログラムの処理速度を検証する。処理速度の測定に用いた PC の構成は表 3.2 に示す。

表 3.2: 実装環境

CPU	Inrwl(R) Cor(TM) i7-3630QM 2.40GHz
RAM	8.00GB
GPU	Intel(R) HD Graphics 4000

パラメータ群として3パターンの処理速度を検証する。第1のものは、暫定的な初期パラメータに設定したパターンである。第2のものは、全てのパラメータを最大値に設定したパターンである。第3のものは、全てのパラメータを最小値に設定したパターンである。各パターンのパラメータの値と検証結果を表3.3に示す。パラメータを最大にしたパターンでも348ミリ秒という結果がでたことから劣化表現を本手法により高速で実現できたと言える。

表 3.3: 速度検証結果

検証項目	暫定値	最大値	最小値
劣化の起こりやすさ $p$	0.6	1.0	0.0
切れ目の幅の細さの度合いの最大値 $m$	20.0	20.0	2.0
切れ目の幅の細さの度合いの最小値 $n$	19.0	19.0	1.0
切れ目の角度のノイズ $z$	4.0	8.0	1.0
切れ目の長さの最大値 $j$	10.0	20.0	2.0
切れ目の長さの最小値 $k$	3.0	19.0	1.0
破れ目の起こりやすさ $d$	10.0	20.0	0.0
1ステップの処理時間(ミリ秒)	342	348	19

以上の結果から、本研究で開発したツールは切れ目と破れ目の劣化をポリゴンの変形によって再現するという手間と時間のかかる工程に対して、自動処理による変形を可能とすることで古紙の3DCGモデリングの制作支援を実現した。

### 3.4 問題点

破れ目の劣化として虫食いを挙げたが、本手法では虫食いなど、紙の周囲以外から劣化を生成することはできない。直方体の周囲に頂点を生成し、その頂点を変形することで行なっており、図形内部に頂点がなく、メッシュ構造となっていない状態である。変形操作は頂点の移動によって行われるので、内部に頂点がない出力したモデルは細かい変形操作が行えない状態になっている。図3.23に、内部に頂点がない出力結果を示す。

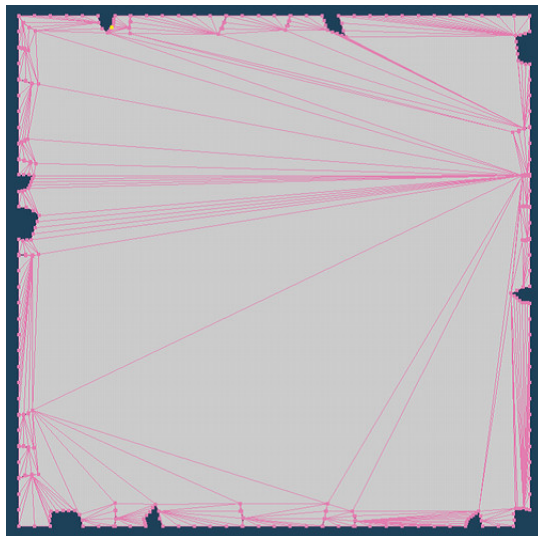


図 3.23: 内部に頂点がない出力結果

また、出力のためにモデルの面全てを三角形化する必要があるが、現状利用している多角形分割システムでは劣化処理を繰り返し行い続けると分割できず、出力できなくなってしまう。これらの問題を解決するためには、劣化処理後に頂点を生成することでメッシュ構造とする方法がある。このことにより、劣化処理をそのままに細かい変形操作を行える構造でかつ、モデルの面全てを三角形化することが比較的容易となる。

加えて、劣化処理はツールに用意したスライダーを操作することである程度 of 出力結果に変化を与えることは可能であるが、処理自体は自動的に行われてしまうためにユーザーの思い描く古紙の劣化が再現できないことが多い。この問題は劣化の発生を乱数に依存し全体に適用していることが原因である。ユーザが劣化させたい箇所を指定できるようにしたり、劣化処理後に一部に細かい変形操作を加えることができるようにすることで、よりユーザが求める劣化を表現したモデリングが可能となる。

## 第 4 章

### まとめ

本手法を用いることで、古紙のモデリングにおいてポリゴン変形による切れ目、破れ目、そして破れ目からの切れ目を作成する工程を自動処理化することができた。このことにより、手間となり時間がかかりやすいポリゴンの変形処理が必要となる工程を大幅に短縮することができ、3DCG コンテンツの制作を効率化することができた。また、形状の変形にあたってツールでスライダーを操作することで多様な切れ目、破れ目を表現することができた。しかし、本研究の出力結果では自由な変形が可能な構造ではないため、より細かい劣化表現をすることはできていない。処理化している劣化の要素も切れ目と破れ目のみであり、しわを含めた湾曲や変色といった経年劣化によるものだけでなく、焦げや濡れといった様々な劣化には対応できない。展望として、出力ファイルをユーザーの手で加工処理しやすいようにメッシュ構造化した上で、切れ目・破れ目を生成できるようになることでよりユーザーの意図する古紙をモデリングすることができるようになる。また、本研究では切れ目・破れ目の劣化表現をポリゴンベースで自動変形処理化することに成功したが、これらは前述の古紙の表現する要素する内の一部でしかない。残りの折れ、しわ、退色といった要素まで自動処理化できるようになれば古紙モデリングの制作は更に効率化が図れることが期待される。

# 謝辞

本研究を行うにあたり多くのご指導をいただきました東京工科大学メディア学部の三上浩司准教授と渡辺大地講師に心より深く感謝いたします。また、竹内亮太演習講師にも多くの助言をいただいたことに深く感謝します。院生の方々には、細かい相談を始めとして幾度も論文校正に時間を割いて頂き深く感謝します。研究を進めるにあたり、相談に乗っていただいた研究室のメンバーに深く感謝いたします。

最後に、人生における活力の源となる音楽ゲームを提供してくれた BEMANI チームと数多くの良曲を提供して BEMANI アーティストに感謝します。

## 参考文献

- [1] コンピュータグラフィックス編集委員会. コンピュータグラフィックス. CG-ARTS 協会, 2004.
- [2] Metasequoia. <http://www.metaseq.net/metaseq/>.
- [3] Maya. <http://usa.autodesk.com/maya/>.
- [4] Blender. <http://blender.jp/>.
- [5] 審良善和 武若耕司 山口明伸 久場公司 . コンクリート構造物の酸性雨劣化シミュレーション. コンクリート工学年次論文集, 2002.
- [6] Julie Dorsey. Alan Edelman, Henrik Wann Jensen, Justin Legekis, Hans Kohling Pederson. Modeling and rendering of weathered stone. *SIGGRAPH '99*, 2006.
- [7] Julie Dorsey. Pat Hanrahan. Modeling and rendering of metallic patinas. *SIGGRAPH '96*, 2006.
- [8] 中島 隆吉. 「紙への道」. <http://homepage2.nifty.com/t-nakajima/>.
- [9] 零 ～真紅の蝶～. コーエーテクモゲームス, 2009.
- [10] FRAGILE ～さよなら月の廃墟～. バンダイナムコゲームス, 2009.



- [11] R. Bruce Arnold. Astm's paper aging research program. <http://cool.conservation-us.org/byauth/arnold/astm-aging-research/>, 2000.
- [12] 大江礼三郎 岡山隆之 松原広明 花村克幸. 紙の劣化とその対策 第 ii 報 機械パ  
ルプ含有紙の老化. 紙パ技協誌, 1991.
- [13] 鈴木英治. 紙の劣化と資料保存. 日本図書館協会, 1993.
- [14] 園田直子 編. 紙と本の保存科学. 岩田書院, 2009.
- [15] 木部徹 鈴木英治. 本の紙の劣化と保存 -歴史に沿って-. CAP 編集室, 1977.
- [16] Best Collection Of Top Quality Free Old Paper Textures. [http://www.  
designresourcebox.com/](http://www.designresourcebox.com/).
- [17] Grenier i-Deco. <http://ideco.ocnk.net/>.
- [18] 佐藤崇雄 藤澤誠 三浦憲二郎. Fem を用いた紙の破断シミュレーション. Visual  
Computing / グラフィクスと CAD 合同シンポジウム 2006, 2006.
- [19] 木谷圭将 田中敏光 佐川雄二. Cg による紙の破断表現. FIT2007 (第 6 回情報  
科学技術フォーラム), 2007.
- [20] 大森健斗. Cg による紙の破れ表現 - 多層繊維分布画像を用いた剥がれ別れ表  
示 -. 電子情報通信学会東海支部卒業研究発表会, 2010.
- [21] 岡敬 芽暁陽 今宮淳美. 紙の劣化のモデリング. 情報処理学会研究報告, 2004.
- [22] 日本設計工学会. 3次元 CAD 実践活用法. コロナ社, 2006.
- [23] 木曾康二 芽暁陽 豊浦正広. 観察に基づく紙の劣化形状モデリングに関する検  
討. 情報処理学会研究報告, 2009.
- [24] 渡辺 大地. 三次元形状のデータ構造と形状変化について. [http://www2.teu.  
ac.jp/aqua/3D/text/PDF/HalfEdge.pdf](http://www2.teu.ac.jp/aqua/3D/text/PDF/HalfEdge.pdf).

[25] 渡辺 大地. Fine Kernel Project. <http://fktoolkit.sourceforge.jp/>.