

修士論文

平成 25 年度 (2013)

ドット絵制作のためのタイルパターングラデーションの
自動生成手法に関する研究

東京工科大学大学院
バイオ・情報メディア研究科 メディアサイエンス専攻

松島 立弥

修士論文

平成 25 年度 (2013)

ドット絵制作のためのタイルパターングラデーションの
自動生成手法に関する研究

指導教員 渡辺 大地

東京工科大学大学院
バイオ・情報メディア研究科 メディアサイエンス専攻

松島 立弥

論文の要旨

論文題目	ドット絵制作のためのタイルパターングラデーションの 自動生成手法に関する研究
執筆者氏名	松島 立弥
指導教員	渡辺 大地
キーワード	ドット絵, タイルパターングラデーション, 画像処理, Self-Quotient Image
[要旨] コンピュータグラフィックスにおいて古くから頻繁に用いられる描写表現として、ドット絵という表現手法が存在する。ドット絵には色数の制限下での表現力を高めるために、さまざまな手法がある。タイルパターングラデーションもその一つであり、独特のドット配置によってグラデーションを再現することが出来る。 しかし、ドットを1つ1つ配置していく作業の綿密さが求められるため、制作には多大な労力と時間がかかってしまう。また、ドット絵制作ツールのタイルパターン機能で対応できない斜線や曲線が境界となる場合は、人の手による修正が必要となる問題がある。 本研究では、ドット絵におけるタイルパターンの表現に着目し、自動でグラデーションを施す手法を提案する。提案手法では、タイルパターンを施す領域生成に Self-Quotient Image を用いた。これにより、ユーザーが入力画像から2色を選択することで、タイルパターン処理を行う領域を自動生成した。その後、各割合のタイルパターンごとに、異なるアルゴリズムでグラデーションの生成を行った。 これらのことより、入力に用いる2つの画像と数値入力だけで、簡単にタイルパターングラデーションを施すことができた。この提案手法を用いて、複数の大きさの画像にタイルパターン処理を実現した。また従来では人の手による修正が必要となる場合でも、グラデーションの境目を明瞭にしつつ、形状に沿った処理結果の取得を実現した。	

A b s t r a c t

Title	The method of automatically generating tile pattern gradation for Pixel Art
Author	Tatsuya Matsushima
Advisor	Taichi Watanabe
Key Words	Pixel art, Tile pattern gradation, Image processing, Self-Quotient Image

[summary]

Pixel art is a description technique of the computer graphic frequently used from of old. There are various techniques to improve the expression to Pixel art by limiting the number of colors. Tile pattern gradation is one of the Pixel art techniques. This technique can reproduce the gradation by peculiar dot arrangement. Tile pattern gradation is used for the expression of texture etc. though the data capacity became unrestricted in several years recently.

However, because the elaborateness of work to arrange the dot is requested, it take a lot of work and time to Tile pattern gradation produce. And the problem that the correction by person's sense is needed exists when the slash and the curve not treatable with the tool become boundaries of the gradation.

In this paper, we focused on the expression of Tile pattern in the Pixel art and suggest technique to automatically generate the gradation. In the proposal method, we used Self-Quotient Image to generate the area of Tile pattern. As a result, the user selected two points from the input image, and the area for Tile pattern was generated with the automatic operation. Afterwards, it generated the gradation by an different algorithm in each Tile pattern of the ratio.

From these, Tile pattern gradation was easily achieved only by two images and one numerical values used for the input. Using this proposal technique, we achieved Tile pattern in the image of two or more sizes. Moreover, the acquisition of the processing result along shape making the verge of the gradation plain even when the correction by person's sense was needed so far was achieved.

目次

第1章	はじめに	1
1.1	研究概要	2
1.2	本論文の構成	7
第2章	ドット絵におけるタイルパターングラデーション	8
第3章	ドット絵のためのタイルパターングラデーションの自動生成手法	11
3.1	提案手法で扱うタイルパターン	12
3.2	提案手法の概要	13
3.3	入力画像の用意と隣接する領域の選択	14
3.4	入力画像と線画画像を用いたラベリング処理	16
3.5	タイルパターン領域 (TPR) の決定	18
3.6	タイルパターンの生成処理	22
第4章	検証と考察	25
4.1	手法の検証	26
4.2	考察	32
第5章	まとめ	34
	謝辞	36
	参考文献	38

目 次

1.1	原寸大のドット絵と4倍に拡大したドット絵の例	2
1.2	dot's による作品例	3
1.3	ドット絵のキャラクターが使用された作品	4
1.4	3D 背景にドット絵のキャラクターが配置された作品	4
1.5	タイルパターングラデーションを用いたドット絵	5
1.6	分散型規則的ディザリングの処理例	6
2.1	階調グラデーションとタイルパターングラデーションの比較画像	9
2.2	EDGE のトーンツールにおけるタイルパターン	10
2.3	曲線状の境界を持つ元画像と、トーンツールを用いた結果と修正画像の例	10
3.1	各割合でのタイルパターンのドット配置例	13
3.2	提案手法の工程	14
3.3	入力画像と対応する線画画像の例	15
3.4	(a, b) を開始位置とした時のラベリング結果	17
3.5	元画像から任意の2点を選択した際のラベリング結果画像	18
3.6	SQI を用いた輪郭抽出の例	19
3.7	SQI を用いた各 TPR の決定手順	20
3.8	50%TPR の決定処理の流れ	21
3.9	25%及び75%TPR の決定結果	22
3.10	50%TPR の色 B のドットを基準とした25%TPR で色 B を配置する位置	23
3.11	最終的に生成する結果画像	24
4.1	ドット絵ツールを使用した際と提案手法を使用した際の結果画像	26
4.2	提案手法と複数色によるグラデーションの比較画像	27
4.3	提案手法の低画素数入力画像への適用結果とその拡大画像	28
4.4	直線状の境界を持つ元画像と、入力値 e を変更した場合の結果画像	29
4.5	直線状の境界を持つ元画像と、入力値 e を変更した場合の結果画像	29
4.6	3D モデルを用いたレンダリング画像と提案手法の適用結果	30

4.7	複数の大きな色領域を持つ元画像と処理結果	31
4.8	複数の小さな色領域を持つ元画像と処理結果	31

第 1 章

はじめに

1.1 研究概要

ビデオゲームを中心に古くから頻繁に用いられるコンピュータグラフィックス (CG) の描写表現として、ドット絵という表現手法が存在する [1][2][3][4]。ドットとは、コンピュータ上におけるディスプレイで表示可能な、最小単位の画素 (Pixel) である。そして、ドットを1つずつ人の手で配置することにより制作した、低画素数ながら素材の概要を視認可能な画像がドット絵である。この際、制作者はドット絵を描くと言うよりも、イメージに即してドットを配置する作業が基本になる。低い解像度でもキャラクターや物体の概要を認識できるようにするため、デフォルメ表現が頻繁に用いられることも大きな特徴である。図 1.1 はディスプレイ上で表示した際の原寸大のドット絵と、視認しやすいように4倍に拡大したドット絵の例である。



図 1.1: 原寸大のドット絵と4倍に拡大したドット絵の例

国外においてのドット絵は Pixel art と呼ばれ、デザイン集として書籍が発行されている [5][6][7]。また図 1.2 のように、ピンとパネルでドット絵を表現できる玩具である “dot's” [8] が存在する。このようにドット絵は、単なる CG にとどまらず芸術表現の1種としても認識が強い。



図 1.2: dot's による作品例

©TOMYTEC

ドット絵は容量制限下で生まれた手法であるが、現在でも携帯ゲーム機やモバイル上のゲームなどの色数や画素が限られた状況では、未だに主要な制作手段として用いられている。綿密な作業によって制作されたドット絵自体の人気もあるため、全盛期が過ぎた今でも表現手法の1種として用いることがある。図 1.3 は“ポケットモンスターブラック2・ホワイト2”[9]でのドット絵キャラクターの例、図 1.4 は“魔界戦記ディスガイア3”[10]で3D背景にドット絵のキャラクターが配置された例である。



図 1.3: ドット絵のキャラクターが使用された作品

©Nintendo/Creatures Inc./GAME FREAK inc.



図 1.4: 3D 背景にドット絵のキャラクターが配置された作品

©Nippon Ichi Software, Inc.

ドット絵制作に関しては、フリーでも扱いやすいドット絵制作ツールも存在し、制作環境は簡単に整う。このことから敷居はむしろ低く、特に近年では、ドット絵専門の制作者は少なく、ドット絵を含め複数の素材制作を行うことが求められ

ている [11]。しかしながら、ドット絵の表現力を高めるには技術と経験が必要となる。よって、制作経験に乏しいグラフィッカーが他の作業と並行して、高品質のドット絵の制作を行うのは困難である。

ドット絵の表現のひとつに、タイルパターングラデーションがある。色数の制限がある環境において、段階的な階調表現を行うための手法である。現在ではフルカラーでのグラデーションも可能になったが、低画素数での凸凹とした質感の再現や、タイルパターン独特の雰囲気などから使用するケースは少なくない [1][2]。図 1.5 はドット絵の質感表現にタイルパターングラデーションを用いた例である。



図 1.5: タイルパターングラデーションを用いたドット絵

作業の綿密さを求められるドット絵の画像制作には、低画素数の作品でも制作に時間がかかってしまう。タイルパターングラデーションに関しても、ドットの配置により階調表現を行うため、通常のグラデーションよりも多大な労力が必要となる。また、タイルパターンによるグラデーション表現は、ドット絵制作の初心者には扱い難い手法であるという問題もある。

ドット絵に関する既存研究では、ドット絵の輪郭線を再現する研究 [12][13] や、ドット絵の制作工程を支援する研究 [14][15]、入力画像をドット絵風に処理する手

法が存在する [16][17][18]。しかし、特徴的なドット配置で表現を行う、タイルパターングラデーションの生成を実現した手法は存在しない。

また、タイルパターンに関しては、印刷に関する技術において関連する手法が存在する。印刷機での出力では階調表現を行う際は、白と黒のドットを効率的に配置する必要がある。この際一定範囲内の点を、パターンを形成して配置することで階調を疑似的に再現する。この技法をディザ法と呼ぶ [19]。

ディザ法の配置にはいくつかのパターンがあり、用途別に使い分けを行う [20]。分散型規則的ディザリングが、ドット絵におけるタイルパターングラデーションの表現に近いと言える。図 1.6 は分散型規則的ディザリングの処理例である。

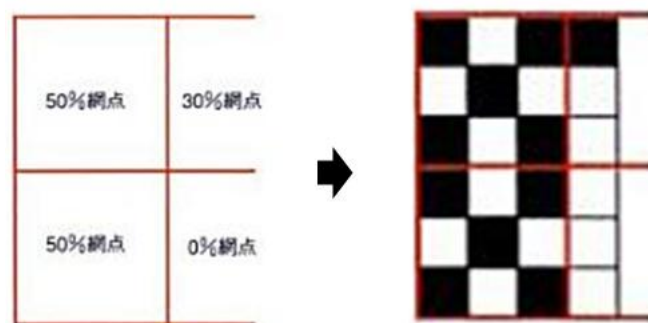


図 1.6: 分散型規則的ディザリングの処理例

ディザ法による配置は、ブロックで区切った範囲ごとに異なる割合のパターンでドット配置を行う。対して、ドット絵は1ドット単位で配置パターンを変えることで、グラデーション表現を行う。このため、ディザ法による階調表現では、ドット絵のような細部を考慮したドット配置を行うことはできない。

ドット絵制作ツールにも、タイルパターンの処理を補助する機能が存在する。フリーのドット絵制作ツール“EDGE” [21] ではトーンツール機能として、一定の画素数において描画色の占める割合により、タイルパターンが段階的に用意されている。制作者はあらかじめ割合ごとに用意された中から任意のパターンを選択することで、タイルパターングラデーションを施すことができる。しかし、グラデー

ションを施す領域はユーザーが自ら描画せねばならず、時間と制作経験が必要となる。また、EDGEの機能を用いた場合は、グラデーションの境界が斜線や曲線となっている際、境界の形状に合わせたドットの配置を行うことができない。これは、トーンツール機能によるタイルパターンが、特定の配置で固定されているためである。このため、グラデーション表現による濃淡に矛盾生じる問題が発生する。加えて、ドット絵は低画素数での表現が可能な利点がある。しかし、画素数が少ないほど1ドットの重要度が増すため、グラデーションの矛盾が目立つ。そのため、タイルパターンを崩すドットが発生した際に、適時削除しなければならない。

本研究ではドット絵制作において表現が困難な、タイルパターングラデーションを自動で処理し効率化を目指す。提案手法では、ユーザーがあらかじめ、元画像と線画画像を入力として用意する。元画像からユーザーが任意の2色の領域を選択することで、選択色と同色の領域にラベリングを行う。ラベリング結果と線画画像を用いることで、各割合のタイルパターン処理を行う領域を決定する。その後、タイルパターン領域画像を参照することで、元画像の対応する領域へ、割合ごとのタイルパターングラデーションを生成した。そして、提案手法が有用であることを証明するため、ドット絵制作ツールを用いたタイルパターングラデーションとの比較検証を行った。その結果、境界の形状に即したドット配置により、タイルパターングラデーションの自動生成を実現した。

1.2 本論文の構成

本論文では、2章にドット絵におけるタイルパターングラデーションの説明を述べ、3章で提案手法に関する説明を述べる。4章で提案手法の検証結果と考察を述べる。5章でまとめを述べる。

第 2 章

ドット絵におけるタイルパターンングラ デーション

ドット絵で用いる独特の表現の1つとしてタイリングがある。色Aと色Bを市松模様のように掛け合わせて中間色の色Cを得る手法がある。ドット絵制作の現場では、このような市松模様での表現はタイリングと呼称する。タイリングに用いる2色を掛け合わせて色相や濃淡を表現し、規則的に変化させたものを、タイルパターンと呼称する [1]。

タイルパターンによるグラデーションも、ドット絵と同じように、色数制限下で多くの色を表現するための手法である。低画素数のドット絵制作やざらつきのある質感、凸凹とした質感の表現を行うため、意図的に用いることがある。タイルパターン自体も独特の雰囲気を持つため、あえて使用する愛好者も少なくない。図2.1はグレースケールの階調グラデーションと、タイルパターンによるグラデーションの比較画像である。

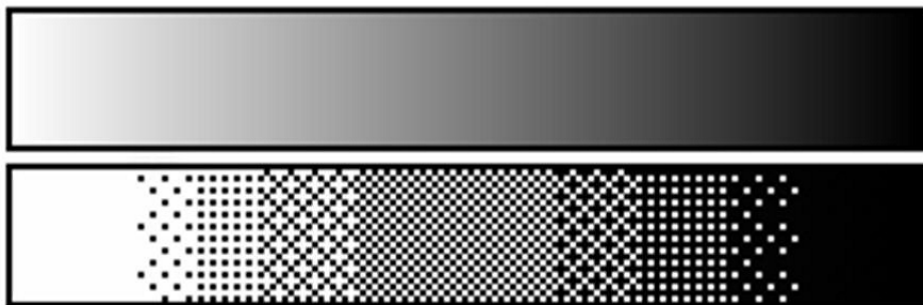


図 2.1: 階調グラデーションとタイルパターングラデーションの比較画像

実際にタイルパターンを施す際は、制作者が1ドットずつ打っていく方法が基本となる。しかし、ドット絵表現では画素数の制限内で意匠性を保つため、単純なパターンではなく形状に即したドット配置が不可欠である。タイルパターンも細部の修正が必要となるため、ドット絵制作経験が乏しい場合には扱いが難しい手法である。

ドット絵制作ツールにも、タイルパターンの処理を補助する機能が存在する。EDGEではトーンツールという機能があり、一定の画素数において描画色の占める割合により、タイルパターンが段階的に用意されている。制作者はあらかじめ

割合ごとに用意された中から任意のパターンを選択することで、タイルパターングラデーションを施すことができる。図 2.2 は EDGE のトーンツール機能で使用可能なタイルパターンである。



図 2.2: EDGE のトーンツールにおけるタイルパターン

トーンツールを用いることで、タイルパターンによるグラデーションの制作がしやすくなる。しかし、トーンツールで塗りつぶしを行う領域は、ユーザーがパターンごとに指定しなければならない。この作業には制作経験が必要であり、時間もかかる。また、グラデーションパターンは特定の配置で固定されており、グラデーションの境界が斜線や曲線となっている場合には対応できていない。図 2.3 は曲線状の境界を持つ元画像へ、EDGE のトーンツールを使用した例である。



図 2.3: 曲線状の境界を持つ元画像と、トーンツールを用いた結果と修正画像の例

図 2.3(b) では、境界部分に周囲のタイルパターンと濃淡の異なる部分が発生している。これにより、本来ならば段階的に濃淡を表現するべきグラデーションに、矛盾が生じる問題が発生する。このような場合は、図 2.3(c) のようにユーザーによる修正が不可欠となるため、制作に掛かる労力と時間が増加する。

第 3 章

ドット絵のためのタイルパターングラ デーシヨンの自動生成手法

本章では、1章で述べた問題点を踏まえ、タイルパターングラデーションを自動生成する手法を提案する。まず、3.1節で提案手法においてタイルパターングラデーションをどのように扱うかについて述べた後、3.2節で提案手法の内容について述べる。3.3節で入力に用いる画像とユーザーの入力について、3.4節でラベリング処理について述べ、3.5節では提案手法で用いる Self-Quotient Image の説明とタイルパターン領域の決定手法について述べる。その後、3.6節でタイルパターングラデーションの生成処理について述べる。

3.1 提案手法で扱うタイルパターン

タイルパターングラデーションはフルカラーでの階調グラデーションと異なり、数段階のドットの配置パターンで表現を行う。しかし、低画素数のドット絵は配置できるドット数の制限から、1ドットごとの重要度が高い。そのため、複数の割合ごとに多くの配置パターンを用いるより、パターン数を限定して適用する。そうすることで、低画素数のドット絵においても、濃淡を認識可能なタイルパターングラデーションを施すことができる。

そこで、本研究では、描画領域に対して描画色の占める割合が25%、50%、75%の3つのタイルパターンで表現する。50%のパターンは市松模様となり、ドット絵でタイルパターングラデーションを用いる場合のもっとも基本的なパターンとなる[1]。また、段階的にタイルパターンを施す際の基準として、グラデーションの中間の位置で使用する人が多いという特徴を持つことから、提案手法での使用を決定した[3]。そのうえで、ドット絵技術書の作例などで使用率の高い、25%および75%となるタイルパターンを使用する。図3.1はそれぞれの割合におけるタイルパターンのドット配置例である。

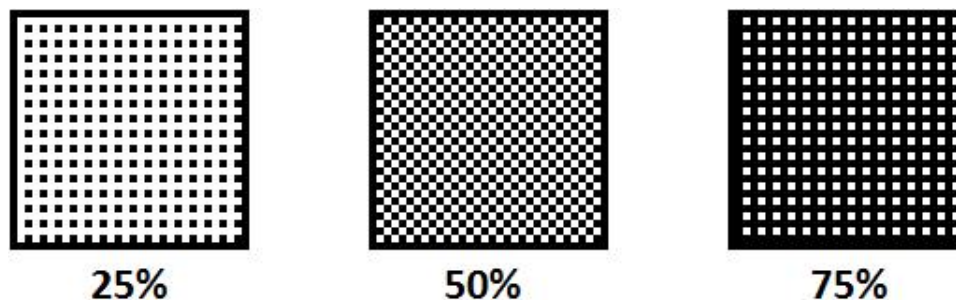


図 3.1: 各割合でのタイルパターンのドット配置例

3.2 提案手法の概要

提案手法では、2つの画像を入力として用いることで処理を行う。この2つの画像は、ユーザーがあらかじめ制作を行い用意する。ユーザーは入力画像から隣接する2色の領域のみを選択し、ぼかしフィルターの大きさを決定する整数値を1つ入力する。残りの操作はプログラムで自動処理する。プログラムでの処理部分では、まずユーザーが選択した2色の領域の境界から、タイルパターンを施す作業領域を生成する。その後、まず50%のタイルパターン領域を決定し、それを元にして25%及び75%のタイルパターン領域の決定処理を行う。図3.2は提案手法の各工程である。

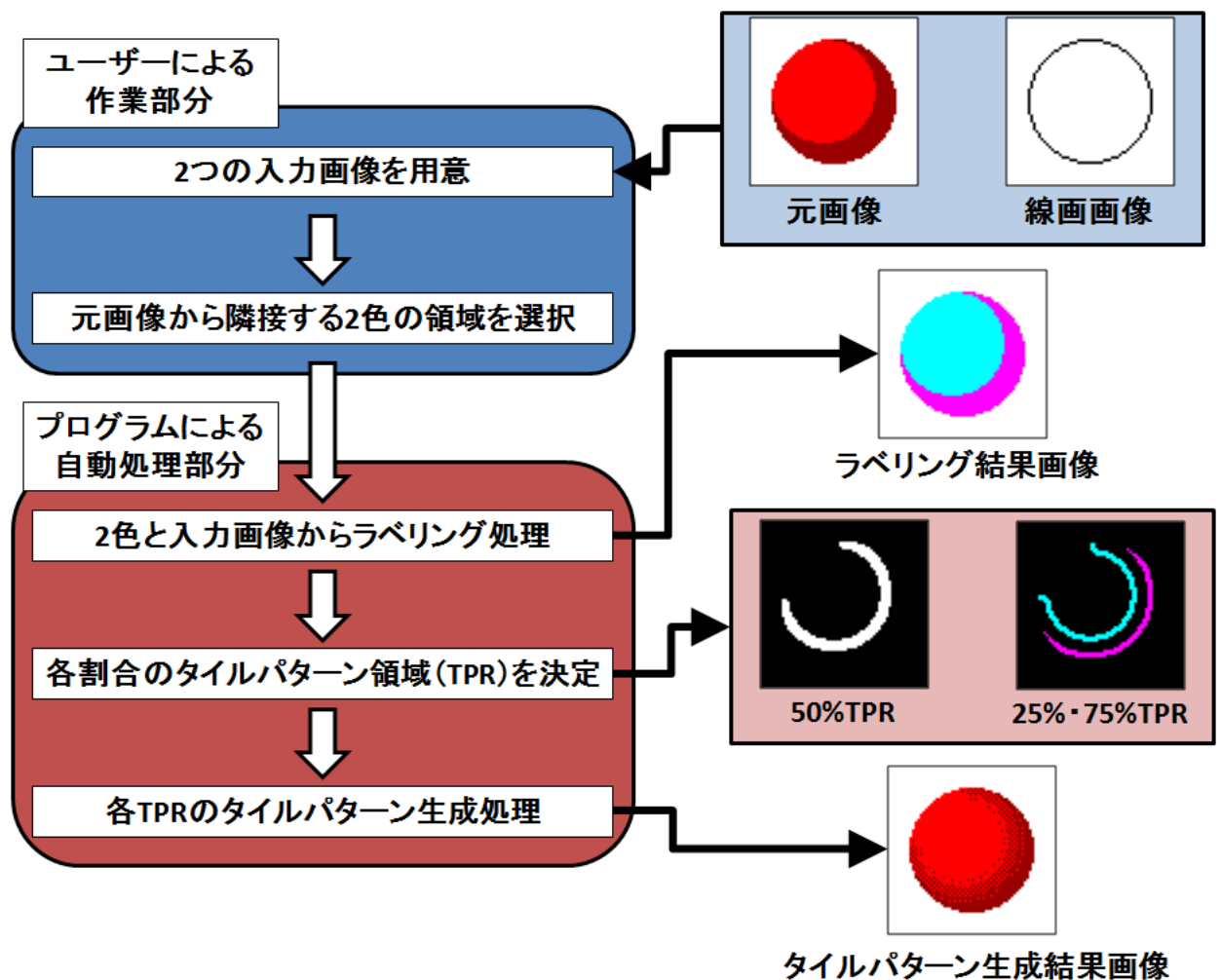


図 3.2: 提案手法の工程

3.3 入力画像の用意と隣接する領域の選択

提案手法では、入力画像として実際にタイルパターン処理を施したい画像を用いる。制作者がこの入力画像から、隣接する2色の領域を任意に選択することで、2色の境界部分にタイルパターンの生成処理を行う。

この際、画像を色による領域選択のみで処理した場合、選択色の隣接する全ての領域にタイルパターン処理を行うことになる。その結果、意図しない位置にグラデーションを生成する問題が発生する。そこで提案手法では、実処理を施す画

像のほかに、処理領域を把握する目的で白黒 2 値の線画画像を用いる。

また、線画画像を用いることで、3D モデルを用いたドット絵制作にも応用が可能となる。3D モデルを元に下絵を描く制作方法の場合も、画像としてレンダリングを行う段階で、3次元での形状情報が失われてしまう問題が発生する。この際、3D モデルの形状把握のために線画画像を別途用意する。これにより、線画で区切った領域に即した任意の部分へ、タイルパターングラデーションを施す事が出来る。線画画像は、基本的に入力画像の輪郭に沿って生成する。そのため、ドット絵の輪郭線の特徴を踏まえることで、より低画素数の入力画像に対応可能となる。図 3.3 は実処理を施す元画像と、各元画像に対応した線画画像の例である。

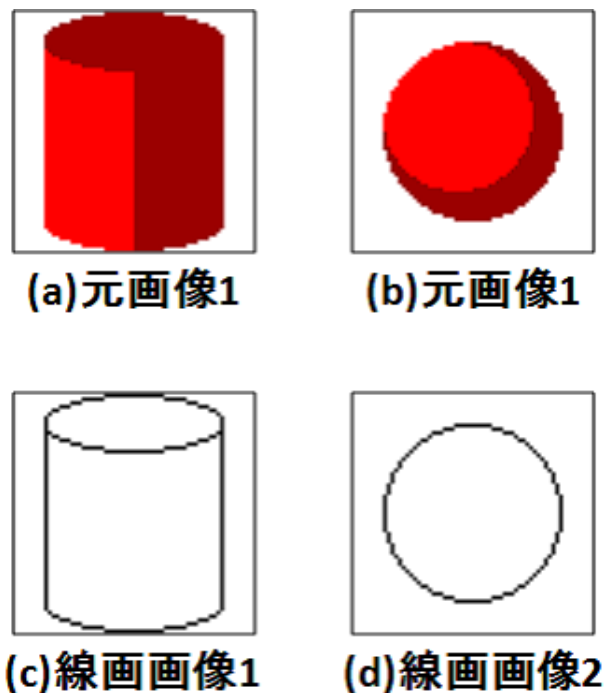


図 3.3: 入力画像と対応する線画画像の例

入力画像を用意したユーザーは、実処理を行う元画像から異なる 2 色を選択し、色 A と色 B を決定する。この 2 色はグラデーションを施すために、1 つの境界を隔てて隣接しており、単色で塗りつぶされた領域に属している必要がある。

3.4 入力画像と線画画像を用いたラベリング処理

2色の領域を選択したのち、選択色を用いてグラデーション処理領域のラベリングを行う。ラベリングとは、画像処理の前段階で行い、図形の領域抽出などに用いる、同じ値を持ちつながっている画素に番号(ラベル)を付ける処理である [22]。ラベリングを行うことで、線画画像で囲まれた領域に即したタイルパターングラデーション処理を行うことが可能になる。

ラベリング処理ではまず、ユーザーが色 A 決定時に選択した位置 (a, b) を元に、線画画像上の同位置からラベリングを開始する。そして、領域拡張法 [23] を用いて、線画で区切った領域内へのラベリングを行う。線画画像上のドットが位置 (x, y) にある時、その位置の線画画像の色を判定する関数 $P(x, y)$ は式 (3.1) のように判定を行う。

$$P(x, y) = \begin{cases} 0 & (\text{ドットの色が黒の場合}) \\ 1 & (\text{ドットの色が白の場合}) \end{cases} \quad (3.1)$$

$P(x, y) = 0$ となる場合、そのドットは線画であるためラベリングできない。 $P(x, y) = 1$ となる場合は、4近傍にラベリング済みのドットが存在する場合、位置 (x, y) にラベリングを行う。この際、ある位置のドットがラベリング済みか判定する関数 $Q(x, y)$ は式 (3.2) のように判定を行う。

$$Q(x, y) = \begin{cases} 0 & (\text{ドットがラベリング済みの場合}) \\ 1 & (\text{ドットにラベリングしていない場合}) \end{cases} \quad (3.2)$$

式 (3.2) により位置 (x, y) のドットは、 $P(x, y) = 1$ かつ式 (3.3) のいずれかの条件を満たす場合に、ラベリングを行う。

$$\begin{cases} Q(x, y - 1) = 0 \\ Q(x + 1, y) = 0 \\ Q(x, y + 1) = 0 \\ Q(x - 1, y) = 0 \end{cases} \quad (3.3)$$

式 (3.1),(3.2),(3.3) を用いて、線画画像上の全てのドットを判別する。これを数回繰り返すことで、線画画像で区切った領域内のラベリングを進めていく。領域内にラベリング可能なドットが無くなった時、処理を終了する。この段階で、線画画像で区切った領域内の、最初にユーザーが選択した 2 色の領域上へのラベリングを施したことになる。図 3.4 はユーザーが色 A を決定する際に選択した位置 (a, b) を開始位置とした時の、線画画像を用いたラベリング結果の例である。

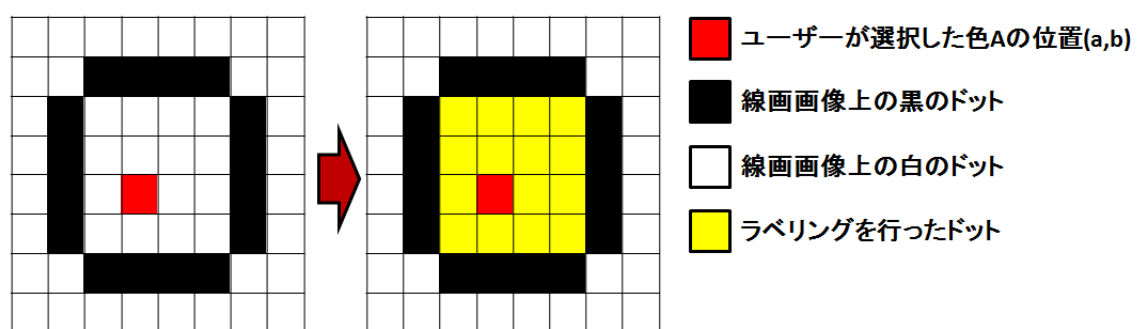


図 3.4: (a, b) を開始位置とした時のラベリング結果

続いて、ラベリング処理済みの各ドットを参照し、元画像上で同座標のドットが色 A であるか色 B であるかを判別する。そのうえで、色 A のラベリング領域、色 B のラベリング領域、ラベリングが施されて居ない領域の 3 つで塗り分けを行い、画像として取得する。図 3.5 は図 3.3(a), 図 3.3(b) それぞれから、2 色の領域を選択した際のラベリング結果画像である。

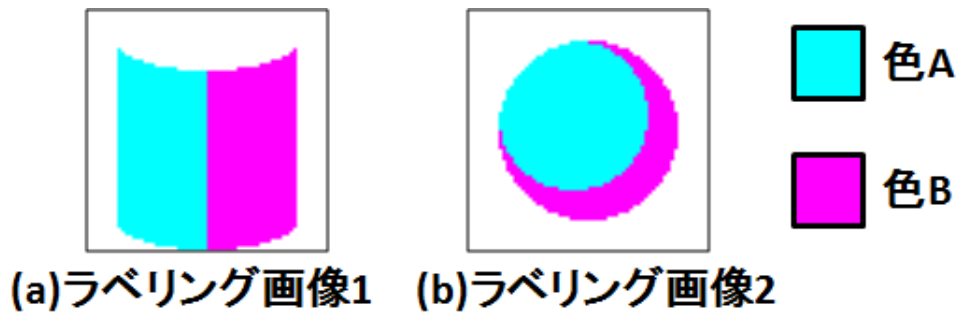


図 3.5: 元画像から任意の 2 点を選択した際のラベリング結果画像

3.5 タイルパターン領域 (TPR) の決定

タイルパターングラデーションの処理には、ドット密度ごとに異なる処理領域が必要となる。本研究において、この各タイルパターンの処理領域を Tile Pattern Region (TPR) と呼称する。提案手法では、タイルパターンを施す領域を自動生成するため、Self-Quotient Image (SQI) の手法を用いる。

SQI は、Wang ら [24] が提案した、写真画像から照明の影響によらずに特徴抽出する為の手法である。Jing ら [25] や菊地ら [26] は、SQI をイラスト風の画像を作成するための輪郭検出に用いた。この結果生成した輪郭線画像を、陰影を塗り分けた画像に重ねることで、実写画像からイラスト風画像への自動処理を実現している。提案手法では、菊地らの手法を元に、SQI を用いた輪郭抽出手法を応用して、TPR の決定処理に用いる。

SQI を用いた輪郭抽出は、処理の元となる画像と、元画像にフィルターをかけてぼかした画像の、2 つのグレースケール画像から行うことができる。ある座標における元画像の画素値を c 、ぼかし画像の画素値を d とする。この時、 c と d の商を用いて 2 値化判定を行う。式 (3.4) となる場合は白とし、式 (3.5) となる場合は黒として 2 値化処理を行う。

$$c \geq d \quad (3.4)$$

$$c < d \quad (3.5)$$

全てのドットにて2値化判定を行い、結果として得られる画像が輪郭抽出の結果画像である。図 3.6 は SQI に用いる元画像と、元画像にフィルターをかけたぼかし画像、2つの画像を用いて生成した結果画像の例である。

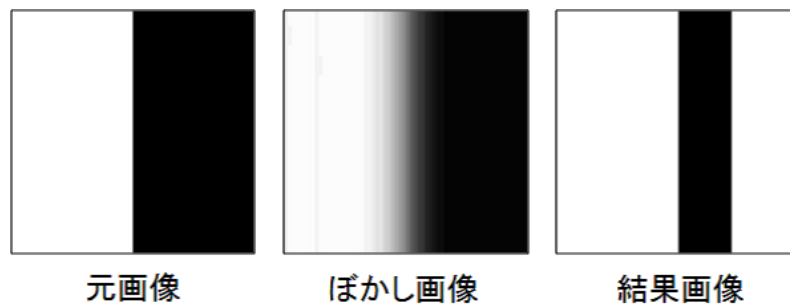


図 3.6: SQI を用いた輪郭抽出の例

SQI は輪郭の検出と輪郭線の描画に用いる手法である。しかし、ぼかし画像生成のためのガウシアンフィルターの半径を決定する任意の整数値 e を入力することで、エッジに沿った任意の太さの輪郭線を自動生成出来る。これを応用することで、グラデーションの生成に必要な色の境界を基準とする処理領域を、少ないユーザー入力により任意の大きさで自動生成する処理が実現可能となる。これにより、ユーザーが TPR を、割合ごとに自力で描画しなければならない問題が解決する。提案手法では SQI を用いることで、境界形状に沿った一定の大きさの領域を取得する。そして、取得した処理結果を組み合わせることで、50%、25%、75%の各 TPR を決定する。図 3.7 は SQI を用いた各 TPR の決定手順である。

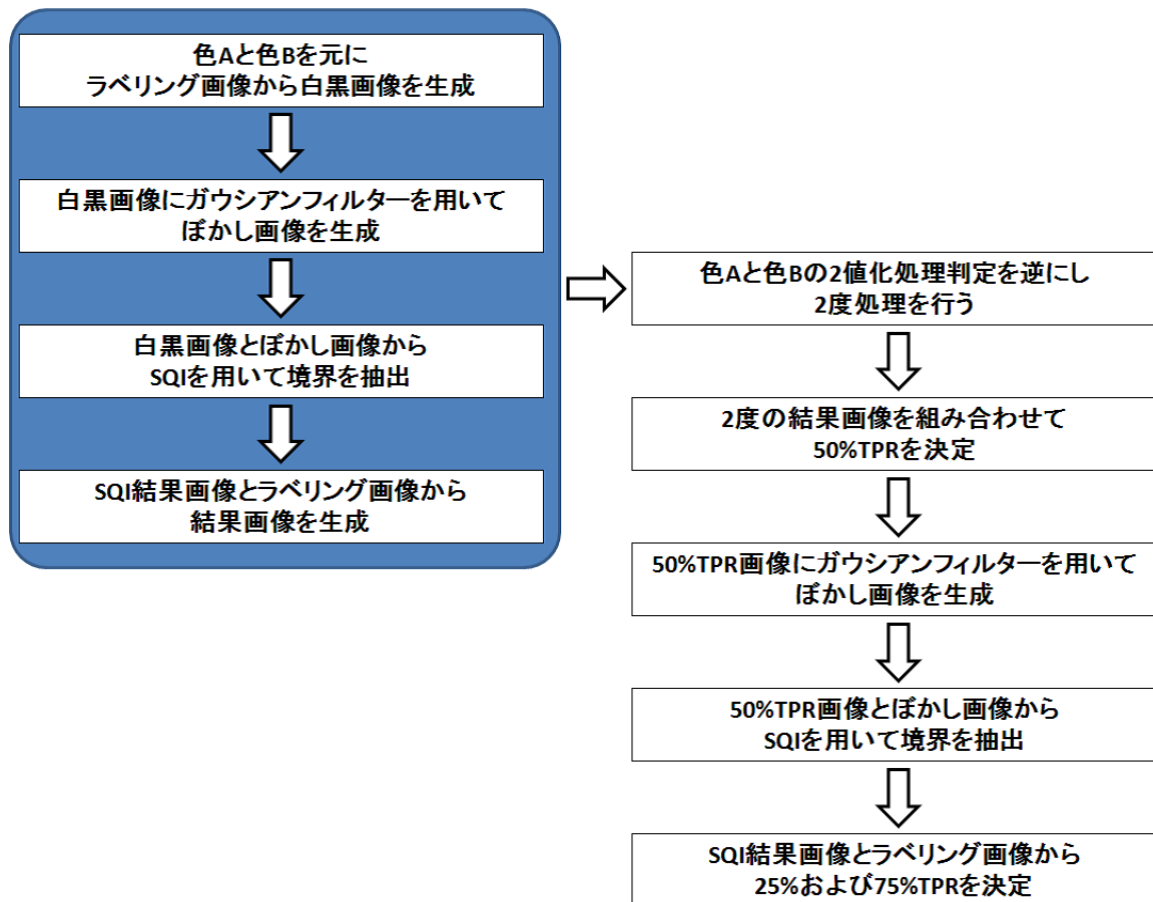


図 3.7: SQI を用いた各 TPR の決定手順

まず、ユーザーが選択した 2 色を元に SQI を用いて、タイルパターングラデーションの基準となる 50%TPR を決定し、画像として保存する。タイルパターングラデーションは任意の 2 色の境界を中間とし、段階的なグラデーションを均等に施す手法である。一方、SQI の処理結果は色の境界を基準として、片方の色の領域上にもみ処理結果を生成する。このため、SQI の処理結果をそのまま用いた場合、色領域に偏りが生じ、結果として TPR が均等に生成出来ない問題が発生する。

そこで提案手法では、まず色 A を白として処理した白黒画像を元に図 3.7 を生成し、ラベリング画像を併用することで、処理領域に即した結果画像を取得する。この際に後の処理を考慮して、SQI の 2 値化処理に変更を加える。白黒画像の画

素値を c 、ぼかし画像の画素値を d とした場合、式 (3.4) となる場合は黒とし、式 (3.5) は白とする。この処理結果から、結果画像を用いた連続的な SQI 処理が可能となる。これにより、50%TPR 画像をそのまま用いて、25%及び75%TPR の決定処理を行うことが出来る。

色 A を白とした SQI 処理結果を取得した後、色 B を白として SQI を用いた 2 値化処理を行う。その後、2 度の処理結果を組み合わせ、50%TPR とすることで、境界を中心とした処理結果を取得可能となる。図 3.8 はユーザーによる任意の入力値 $e = 2$ とした際の、処理の流れと結果画像の例である。

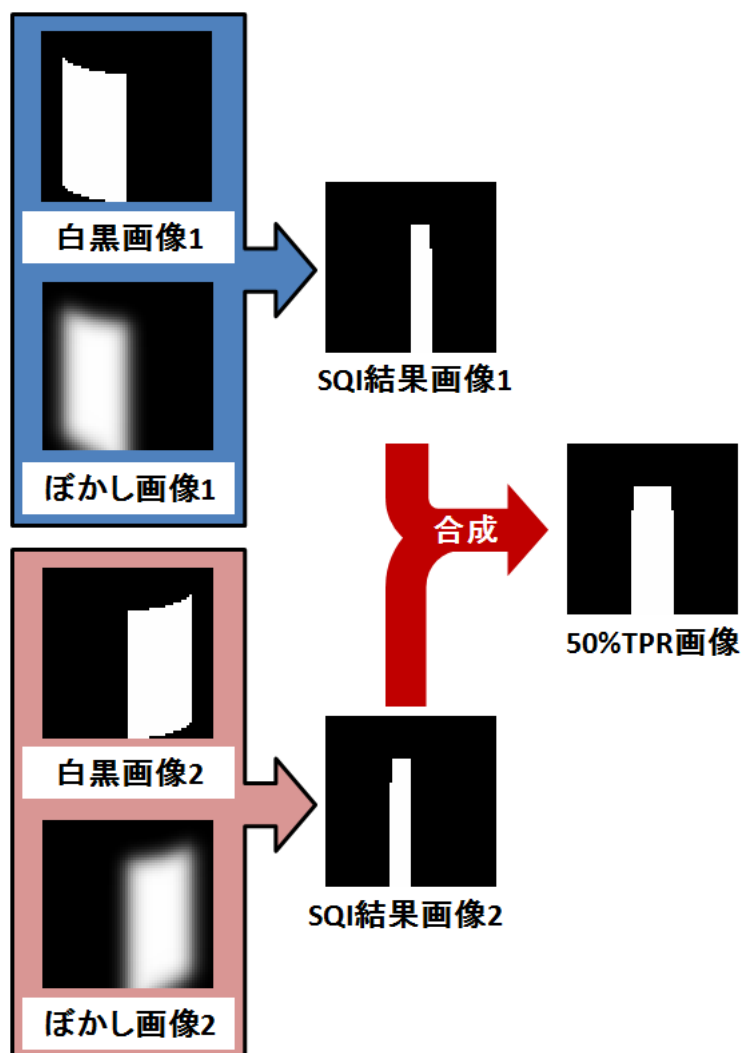


図 3.8: 50%TPR の決定処理の流れ

続いて、50%TPR の画像を元に、再び SQI 処理を行う。先述した 2 値化処理結果を用いることで、50%TPR と重複することなく SQI 処理を行うことができる。その後 SQI の処理結果による 2 値画像に、さらにラベリング画像を用いて 25%及び 75%TPR を決定する。図 3.5 のように、ラベリング画像には判定に用いたドットの色によって、塗り分けが行われている。これにより、色 A に対応するラベリング領域に属した場合は 25%、色 B に対応する場合は 75%として、それぞれ処理し TPR 画像を作成する。図 3.9 は図 3.3(a), 図 3.3(b) それぞれの、25%及び 75%TPR 決定処理の結果画像である。

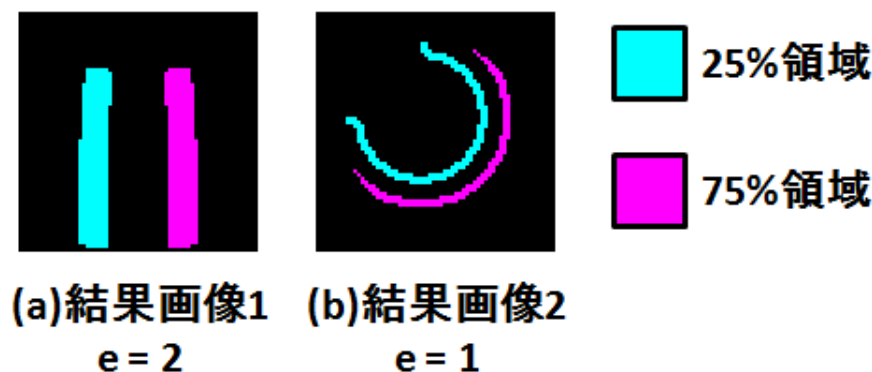


図 3.9: 25%及び 75%TPR の決定結果

3.6 タイルパターンの生成処理

3.5 節で生成した TPR に合わせて、各タイルパターンの生成処理を行う。まず、50%TPR にて市松模様となるように、色 A と色 B の配置を行う。ここでは、TPR 画像を参照した際に 50%TPR に属する、あるドットの位置を (i, j) とする。この時、 $(i + j) \bmod 2$ の計算結果の余りが 0 となる場合、元画像において位置 (i, j) のドットを色 A で置きかえる。同様に、余りが 1 となるドットは色 B で置きかえる。

続いて、25%TPR での処理を行う。処理前の 25%TPR は色 A のみ存在するため、色 B のドットをアルゴリズムにて配置していく。しかし 1 章の図 2.3 のように、斜線や曲線へ単一の配置パターンで処理を行った場合、グラデーションの濃

淡に矛盾を生じるドット配置を行う問題が発生する。そこで25%TPRでのドット配置の基準となるドットを決定する。この際、50%TPR内の色Bのドットの位置を (p, q) 、25%TPR内の参照する位置を (r, s) とするとき、式(3.6),(3.7)の条件を満たすこととする。

$$r = \begin{cases} p & (s \neq q \text{ のときのみ}) \\ p \pm 2 & \end{cases} \quad (3.6)$$

$$s = \begin{cases} q & (r \neq p \text{ のときのみ}) \\ q \pm 2 & \end{cases} \quad (3.7)$$

これにより、25%TPR内に配置した色Bが原因となり、グラデーションに矛盾を生じるドット配置を避ける。図は50%TPR内の色Bのドットを基準とした場合に、25%TPRで配置する色Bの位置を示した例である。

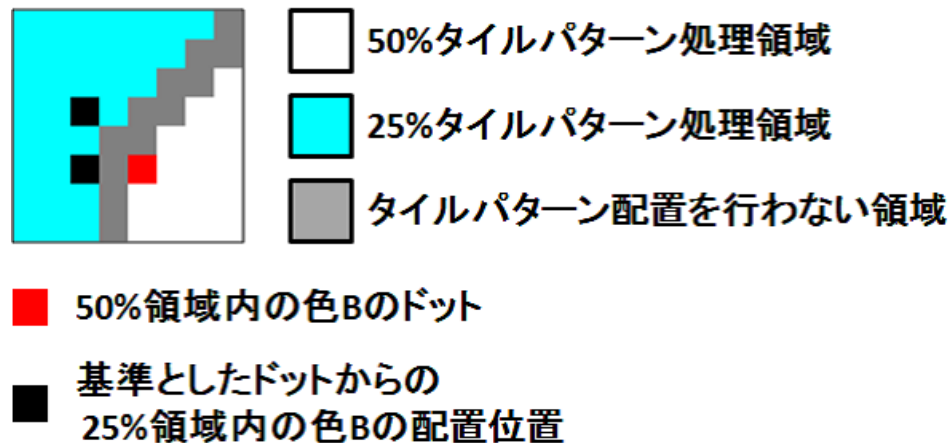


図 3.10: 50%TPR の色 B のドットを基準とした 25%TPR で色 B を配置する位置

その後、境界に沿って25%TPR内に配置した色Bのドットを基準として、再度25%TPR内のタイルパターン処理を進める。この際は、すでに25%TPR内に存在する色Bのドットの位置を (k, l) 、25%TPR内の色Aのあるドットの位置を (m, n) とした場合、式(3.8),(3.9)の条件を満たす位置のドットを色Bで置きかえていく。

$$m = \begin{cases} k & (n \neq l \text{ のときのみ}) \\ k \pm 2 & \end{cases} \quad (3.8)$$

$$n = \begin{cases} l & (m \neq k \text{ のときのみ}) \\ l \pm 2 & \end{cases} \quad (3.9)$$

しかし、式 (3.8),(3.9) の条件のみでドット配置処理を行うと、25%TPR 内で配置した色 B のドット同士が 8 近傍に隣接し、タイルパターンの割合が 25%以上になる箇所が生じる問題が発生する。そのため、25%TPR 内のある色 B のドットの位置を (k, l) 参照し、タイルパターンの修正処理を行う。この時、式 (3.10),(3.11) の条件を満たす、25%TPR 内において位置 (m, n) の色 B のドットを、色 A で置きかえる。これにより、割合以上に密になったタイルパターンを修正する。

$$m = \begin{cases} k & (n \neq l \text{ のときのみ}) \\ k \pm 1 & \end{cases} \quad (3.10)$$

$$n = \begin{cases} l & (m \neq k \text{ のときのみ}) \\ l \pm 1 & \end{cases} \quad (3.11)$$

75%TPR の場合も、色 B ではなく色 A を基準として、25%TPR と同様にドットを配置していく。図 3.11 は、図 3.3(a) と図 3.3(b) を元画像として用いた際に、最終的に生成した結果画像である。

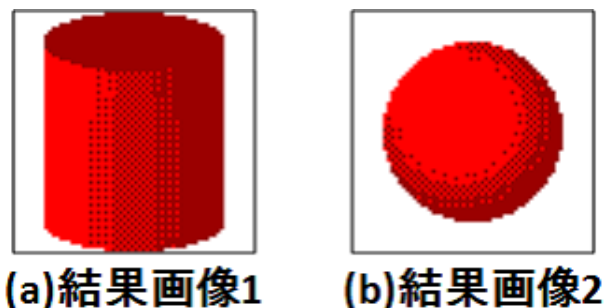


図 3.11: 最終的に生成する結果画像

第 4 章

検証と考察

本章では、3章で述べた提案手法の検証を行う。4.1節にて提案手法の検証を行い、4.2で考察を述べる。

4.1 手法の検証

検証に用いたプログラムはオープンソースのプログラミング言語である“Processing[27]”を用いて実装した。図4.1は入力画像のサイズが 64×64 、ユーザーの入力値 $e = 1$ とした場合の、実処理を行う前の元画像と、2点を選択した結果決定した25%、50%、75%TPRの画像、ドット絵制作ツール“EDGE”のトーンツールを用いたタイルパターングラデーションの画像、及び本研究の提案手法を用いた結果画像である。

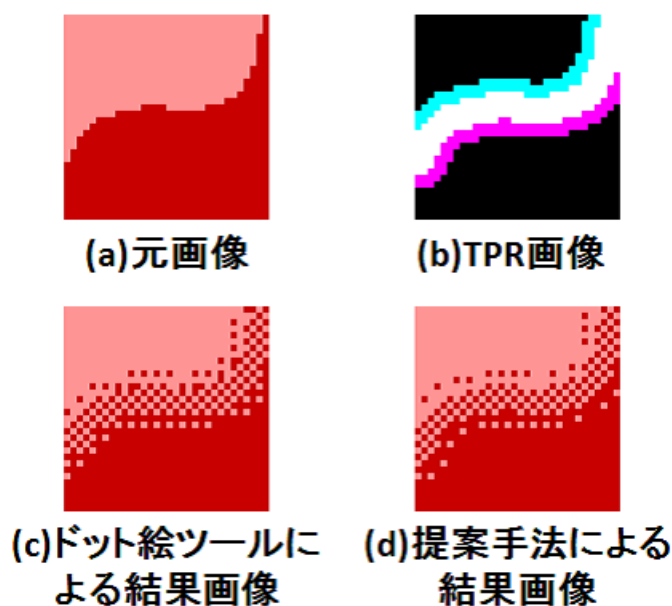


図4.1: ドット絵ツールを使用した際と提案手法を使用した際の結果画像

図4.1(a)が入力に用いた画像、図4.1(b)がTPR画像、図4.1(c)がTPR画像を元にEDGEのトーンツール機能でタイルパターンを生成した画像、図4.1(d)がTPR画像を元に入力画像へタイルパターン自動生成した画像である。

また、元画像へのタイルパターングラデーションと、複数の色を用いたカラーグラデーションとの比較も行った。図 4.2 は入力画像のサイズが 64×64 、入力値 $e = 3$ とした場合の、入力に用いた画像と提案手法によるタイルパターン結果、同じ元画像を用いて複数の色によるグラデーションを施した画像の比較例である。

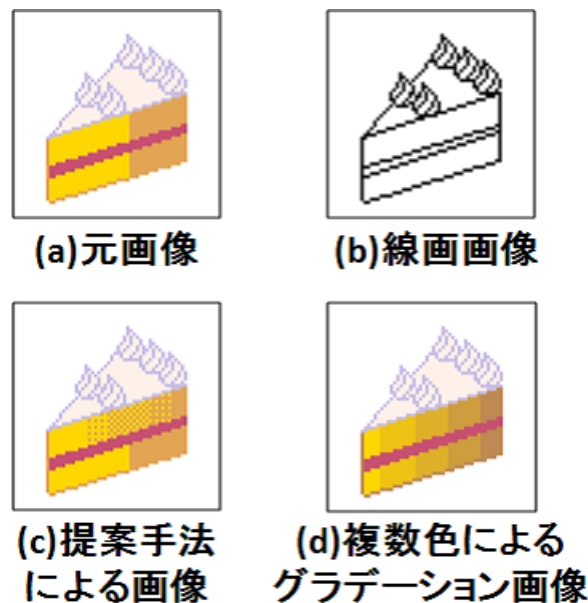


図 4.2: 提案手法と複数色によるグラデーションの比較画像

図 4.2(a) が元画像、図 4.2(b) が線画画像である。図 4.2(c) が提案手法を用いて生成した結果画像であり、図 4.2(d) が元画像に複数色での階調グラデーションを施した画像である。

ドット絵は低画素数の画像である。そのため提案手法は、低画素数のドット絵にも対応可能である必要がある。そこで、小さな画像サイズと色領域の元画像に対して、提案手法を用いてタイルパターングラデーション処理を施し、検証を行った。図 4.3 は入力値 $e = 1$ とした、複数のドット絵技術書 [3][4] で用いられている画像サイズが 32×32 の入力画像及び処理結果画像と、その拡大表示画像である。



図 4.3: 提案手法の低画素数入力画像への適用結果とその拡大画像

図 4.3(a) は 2 色の領域の境界が斜線状である低画素数の元画像、図 4.3(e) は曲線状の領域を持つ低画素数の元画像である。図 4.3(b), 図 4.3(f) がそれぞれの画像に対応する線画画像、図 4.3(c), 図 4.3(g) がタイルパターンの領域画像、図 4.3(d), 図 4.3(h) が最終的な結果画像である。

SQI を用いた手法では、入力値 e に任意の整数値を設定することで、多様な大きさの領域を得ることができる。そこで、提案手法でも境界形状に即した領域を、入力値 e を変更することで取得可能か検証を行った。図 4.4 は画像のサイズが 64×128 で、2 色の領域間に直線状の境界形状を持つ元画像と、入力値 e を変更した場合の結果画像である。また、図 4.5 は画像のサイズが 128×128 で、2 色の領域間に曲線状の境界形状を持つ元画像と、入力値 e を変更した場合の結果画像である。

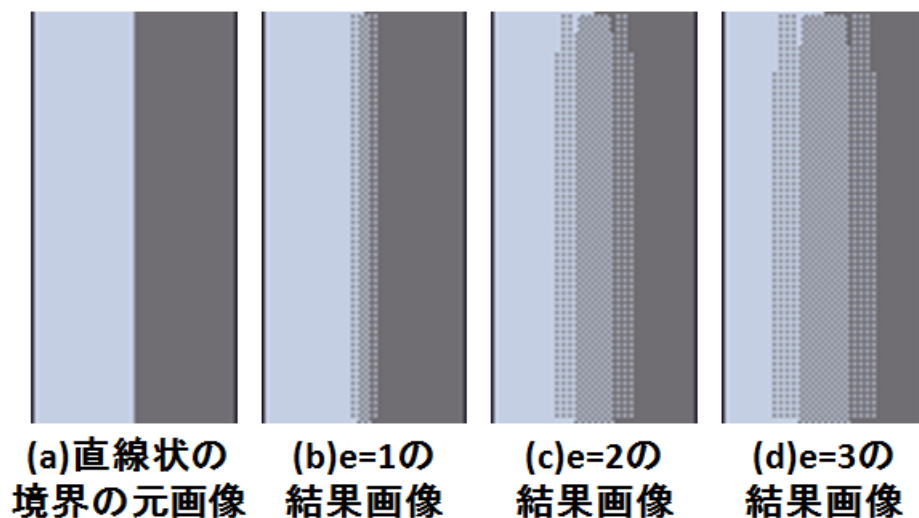


図 4.4: 直線状の境界を持つ元画像と、入力値 e を変更した場合の結果画像

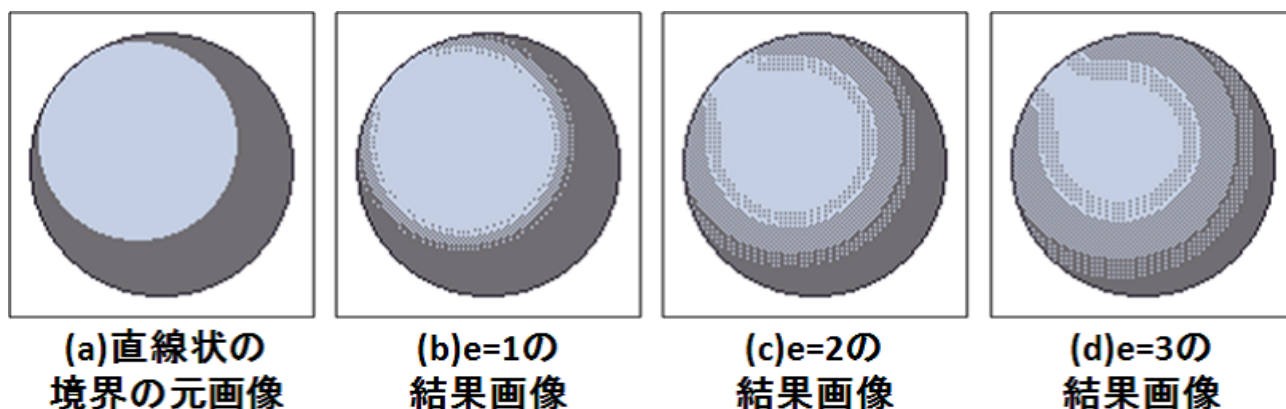


図 4.5: 直線状の境界を持つ元画像と、入力値 e を変更した場合の結果画像

図 4.4(a), 図 4.5(a) がそれぞれの元画像であり、図 4.4(b), 図 4.5(b) が入力値 $e = 1$ の時、図 4.4(c), 図 4.5(c) が入力値 $e = 2$ の時、図 4.4(d), 図 4.5(d) が入力値 $e = 3$ の時の提案手法を用いた結果画像である。

さらに、提案手法が 3D モデルを用いたドット絵制作手法にも応用可能であるか検証を行った。検証にあたり、元画像として 3D モデルに Non-Photorealistic Rendering(NPR)[28] 処理を行った画像を使用する。レンダリング結果は、明部と

暗部の2色の陰影で構成された画像として取得する。3Dモデルのレンダリング用プログラムは、グラフィックスAPIの“OpenGL[29]”とOpenGL拡張機能のライブラリである“OpenGL Extension Wrangler Library[30]”、及びシェーディング言語の“OpenGL Shading Language[31]”を用いて実装している。入力用線画はレンダリング画像を下絵として、ペイントツールを用いて制作した。図4.6は3Dモデルを用いたレンダリング結果画像と、レンダリング結果を減色した比較画像、及び提案手法を適用した結果画像である。



図 4.6: 3D モデルを用いたレンダリング画像と提案手法の適用結果

図 4.6(a) は 3D モデルのフルカラーでのレンダリング結果であり、図 4.6(b) では図 4.6(a) を減色した結果画像である。図 4.6(c) は NPR により、2 階調の陰影を

持つレンダリングを行った結果画像である。図 4.6(d) は図 4.6(c) を元画像として、3D モデルの側面にあたる部分への、タイルパターングラデーション処理を実現している。

さらに、線画画像で区切った範囲内において、2 色かつ複数の色領域を持つ元画像へ、提案手法を用いた検証を行った。図 4.7 は入力値 $e = 1$ で、複数の大きな色領域を持つ元画像と提案手法による処理結果であり、図 4.8 は複数の小さな色領域を持つ元画像と提案手法による処理結果である。



図 4.7: 複数の大きな色領域を持つ元画像と処理結果

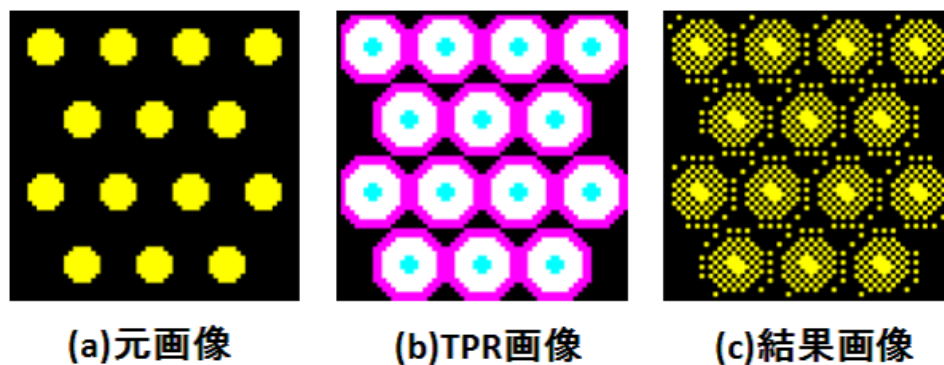


図 4.8: 複数の小さな色領域を持つ元画像と処理結果

図 4.7(a) は 2 色かつ複数の大きな領域を持つ元画像、図 4.8(a) は 2 色かつ複数の小さな領域で構成した元画像である。図 4.7(b)、図 4.8(b) はそれぞれの元画像が

ら決定した TPR 画像であり、図 4.7(c), 図 4.8(c) は元画像と TPR 画像から生成した結果画像である。

4.2 考察

本研究の手法は、タイルパターングラデーション処理を行う領域の生成と、2色の領域における境界の形状を問わないドット配置を、自動処理で行うために提案し、検証を行った。

図 4.1(b) の TPR 画像を見ると、提案手法を用いることで、タイルパターンを施す領域の自動生成を実現している。また、提案手法による結果画像である図 4.1(d) は、ドット絵ツールを使用した結果である図 4.1(c) において各 TPR の境界に発生していた、タイルパターングラデーションの濃淡に矛盾を生じるドットが存在していないことが判る。また、図 4.1(c) では 75%TPR に当たる部分のタイルパターンにおいて、ドットの配置の連続性が途切れている部分が存在する。このため、どこまでが 50%のタイルパターンであり、どこまでが 75%のタイルパターンであるかの認識が困難になる問題が発生している。対して図 4.1(d) はほぼ一定の間隔で、タイルパターンを構成するドットを配置できている。これにより、提案手法は領域の形状に沿ったタイルパターンを自動生成出来たと言える。

図 4.2 では、図 4.2(a) の元画像と図 4.2(b) の線画画像から提案手法を用いて生成した図 4.2(c) と、図 4.2(a) の元画像にフルカラーの階調グラデーションを施した図 4.2(d) の比較を行った。タイルパターンを用いたグラデーション表現が、フルカラーでの階調グラデーションよりも、ケーキのスポンジの凸凹とした質感を表現できていることが判る。

図 4.3 では、低画素数のドット絵を入力に用いた際も、2色の領域の境界形状に即した結果を得ることができた。また、図 4.4(b) や図 4.5(b) では、入力値 $e = 1$ の場合に、高画素数を元画像においても、境界形状に沿ったタイルパターンの生成ができている。しかし、図 4.4(c) や図 4.4(b) では、 $e = 2$ 以上とした場合に TPR の上部分に領域が狭くなる箇所が発生している。また、図 4.4(c) や図 4.4(b) では、

$e = 2$ 以上とした際に、各画像の左上の鋭角となっている 2 色間の境界部分が、 e の値が大きくなるにつれて丸みを帯びていく問題が発生している。これは、SQI に用いるばかり画像生成処理において、ガウシアンフィルターの形状による影響が出たためである。

また、図 4.7 と図 4.8 の結果にて、線画で区切った領域内に 2 色かつ複数の領域で構成した元画像へ提案手法を適用可能であるかを検証している。図 4.7(c) において、ある程度の大きさの領域間であれば、複数の領域の境界間にもタイルパターンを施すことができている。しかし、図 4.8(c) では、領域全てに SQI を用いた結果、過剰にタイルパターンを生成する問題が発生した。このため、今後さまざまな画素数のドット絵に対応するためには、SQI を用いた TPR 決定処理のさらなる改良が必要である。

第 5 章

まとめ

本研究では、タイルパターングラデーションを自動生成するための手法を提案し、検証を行った。入力に用いる2つの画像と数値入力だけで、任意の2色の領域間にタイルパターングラデーションを施すことができた。また、従来では境界が斜線や曲線の場合など、グラデーションの濃淡に矛盾を生じる問題が発生した。このような場合に、人の手によるタイルパターンの修正が不可欠となり、制作時間と労力が必要であった。これに対し、タイルパターングラデーションの意匠を保ちつつ、形状に沿った処理結果を得ることで、問題を解消することができた。

本研究ではタイルパターン処理を、各ドットの色を条件ごとに置きかえるという、単純な処理によって実現している。そのため、複数の割合のタイルパターンを用いての処理が難しい。また、使用するタイルパターンの割合ごとに、改めてアルゴリズムを検証しなければならない。各TPRのタイルパターン処理を最適化することで、より多くの割合のタイルパターン用いたグラデーションの自動生成の実現が望める。

近年ではタイルパターングラデーションは様々な場所で用いられるようになってきている。たとえば電光掲示板にLEDが使用することで、多様な表現が可能となっている[32]。LEDが表現できる色数には限りがあるため[33]、タイルパターングラデーションを取り入れることで、より表現の幅を広げることができる可能性がある。今後はタイルパターングラデーションを、ドット絵の表現手法としてだけでなく、さまざまな表現に活用できるように模索していきたい。

謝辭

本論文を締めくくるにあたり、大学院での研究生生活の間、指導教員として長くに渡り的確なご指導とご助言をくださいました渡辺大地先生、学部1年次のプロジェクト演習から実に9年もの間、多くのご指導をくださいました三上浩司先生に、深く感謝致します。在学期間が大変長くなってしまいましたが、本当にお世話になりました。ドット絵関連研究の情報とご助言をくださり、副査お受けいただいた近藤邦夫先生、進捗報告や研究相談の際に、多くのご助言をくださった柿本正憲先生にも、感謝の意を表します。そして、論文や研究内容へのアドバイスをくださり、休学の後に戻ってきた際も温かく迎えてくれた歴代の院生、学部生の皆様。苦しい時や悩んだ時にいつも励ましてくれた、プロジェクト演習やサークルの先輩、同期、後輩の皆様。研究生生活と療養生活をさまざまな面から、いつも心配し支えてくれた家族に、厚くお礼を申し上げます。

参考文献

- [1] 高野 隼人. ドット絵プロフェッショナルテクニック - ドット打ちからアニメーションまで. 秀和システム, 2005.
- [2] 瑞穂 わか, 中村 竜彦. GameGraphicsDesign ドット絵キャラの描き方. ソフトバンククリエイティブ, 2008.
- [3] Suguru.T. ドット絵職人 - すべてのパソコンに入っている「ペイント」ツールでつくる. エムディエヌコーポレーション, 2007.
- [4] 鯖戸 昭史. ドット絵工房 - 誰でも使えるドット絵テクニックが満載!! エクスメディア, 2006.
- [5] IdN. *Pixel World*. Laurence King Publishing, 2003.
- [6] Sebastian Mendez, Vasava Artworks. 1x1: Pixel-Based Illustration & Design. Mark Batty Pub, 2004.
- [7] Pepin Van Roojen. *Mini Icons (Agile Rabbit Editions)*. Pepin Pr, 2003.
- [8] TOMYTEC. dot 's. http://www.tomytec.co.jp/item/item_dots.html.
- [9] Nintendo, Creatures Inc. GAME FREAK inc. ポケットモンスターブラック2・ホワイト2. <http://www.pokemon.co.jp/ex/b2w2/>.
- [10] Nippon Ichi Software, Inc. 魔界戦記ディスガイア3. <http://disgaea.jp/3/>.
- [11] 成美堂出版編集部. ゲームクリエイターになりたい人の本 - ゲーム業界で働きたい! 成美堂出版, 2008.
- [12] 鈴木和明, 近藤邦雄. ドット絵の分析とドット絵のための輪郭線描画・縮小手法. 画像電子学会, Vol. 219, pp. 35-42, 2005.
- [13] Tiffany C. Inglis, Craig S. Kaplan. Pixelating vector line art. *SIGGRAPH '12*, 2012.

- [14] 松島立弥, 三上浩司. シームカービング手法を用いた人物ドット絵制作の効率化手法. 情報科学技術フォーラム講演論文集, Vol. 8, No. 3, pp. 293–294, 2009.
- [15] 毛利麻美, 三上浩司. 3d モーションデータを用いたドット絵制作のための動きガイド自動生成. 東京工科大学 メディア学部 コンテンツプロデュースング, 2012.
- [16] 伊藤 庸一, アンドレア・クティチネー・マツ. ラスタ型画像への自動アンチエイリアシングアルゴリズムの開発. 東京工科大学 メディア学部 イメージメディア処理プロジェクト, 2006.
- [17] Johannes Kopf, Ariel Shamir, Pieter Peers. Content-adaptive image down-scaling. *Transactions on Graphics*, Vol. 32, , 2013.
- [18] Inglis, Tiffany C, Vogel, Daniel, Kaplan, Craig S. Rasterizing and antialiasing vector line art in the pixel art style. In *NPAR '13*, pp. 25–32, 2013.
- [19] 澤田 善彦, 玉虫 幸雄. DTP エキスパート用語辞典. 日本印刷技術協会, 2002.
- [20] 東 陽一. DTP テクニカルキーワード 250 カラー図解. 日本印刷技術協会, 2005.
- [21] 飯田 貴士. EDGE. <http://www.tkb-soft.hmcbest.com/index.htm>.
- [22] 酒井 幸市. 画像処理とパターン認識入門 - 基礎から VC#/VC++ .NET によるプロジェクト作成まで. 森北出版, 2006.
- [23] 精密工業会画像応用技術専門委員会. 画像処理応用システム: 基礎から応用まで. 東京電機大学出版局, 2000.
- [24] Yangsheng Wang Jianjun Zhang Haitao Wang, Stan Z. Li. Self quotient image for face recognition. *IEEE*, 2004.

- [25] Kiichi Urahama Jing Linlin, Kohei Inoue. Generating illustration images with isoluminant colors. *IWAIT*, 2006.
- [26] 菊地 翔太, 豊田 敬央, 宮岡 伸一郎. 実写動画像のイラスト風変換処理. 情報処理学会第 72 回全国大会, pp. 2.649–2.650, 2010.
- [27] Ben Fry, Casey Reas. Processing.org. <http://www.processing.org/>.
- [28] Thomas Strothotte, Stefan Schlechtweg. *Non-Photorealistic Computer Graphics: Modeling, Rendering, and Animation*. Elsevier, 2002.
- [29] Silicon Graphics International Corp. java.com. <http://www.opengl.org/>.
- [30] Milan Ikits, Marcelo Magallon. The OpenGL Extension Wrangler Library. <http://glew.sourceforge.net/>.
- [31] The Khronos Group Inc. The OpenGL Shading Language. <http://www.opengl.org/documentation/glsl/>.
- [32] 組込みプレス Vol.18. 株式会社 技術評論社, 2010.
- [33] 長谷川竜生, 釜野勝, 上原信知. 図解入門よくわかる最新 LED の基本と仕組み. 秀和システム, 2012.