

洋服シワに対するキャラクターイラスト特有の  
影形状のリアルタイムレンダリング

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

青木 明優花

洋服シワに対するキャラクターイラスト特有の  
影形状のリアルタイムレンダリング

指導教員 渡辺 大地 准教授

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

青木 明優花

## 論文の要旨

|       |                                       |
|-------|---------------------------------------|
| 論文題目  | 洋服シワに対するキャラクターイラスト特有の影形状のリアルタイムレンダリング |
| 執筆者氏名 | 青木 明優花                                |
| 指導教員  | 渡辺 大地 准教授                             |
| キーワード | イラスト、レンダリング、陰影、シェーダ、シワ                |

### [要旨]

近年、コンピュータを用いてイラストを描く人が増加しており、ツールの機能向上と描画手法の発展からクオリティの高いイラストを仕上げることができるようになった。イラストのクオリティを上げる重要なポイントはいくつかあるが、そのポイントの1つに陰影がある。陰影付けの作業には慣れが必要であり、仮に作業に慣れたとしても作業時間や負荷が大きいという問題がある。キャラクターの陰影付けでは、様々なパーツに対して陰影付けを行うが、その中でも洋服は大部分を占め重要である。さらに洋服の陰影付けではシワの影響が大きい。イラストでは、物理法則に従わない陰影の付け方をすることがあり、影の形状はイラスト制作者によっても異なる。そこで本研究ではイラストへの陰影付けの作業時間や負荷の軽減を目的とし、洋服シワに対してできるイラスト特有の影の生成手法を提案する。

イラストにおいてシワは、見栄え良く見えるように記号化した線で表現している。陰影の付き方は、シワが示している意味によって変わる。そこで本研究ではシワ線を「稜線シワ」と「輪郭シワ」の2種類に分類した。

シワ線を「稜線シワ」と「輪郭シワ」の、それぞれに対して影計算でのアルゴリズムを変更することで、陰影を生成する。まずシワ情報を入力する。そこから法線マップと距離マップを作成する。そして陰影計算では、作成したマップを使用してシェーダーで陰影計算を行う。稜線シワは法線マップを使用し、輪郭シワは距離マップを使用して計算を行う。最後に計算結果の陰影の情報を組み合わせて陰影を生成する。

これらの手法を実装し、洋服シワに対するイラスト特有の陰影付けを実現した。法線マップを使用したレンダリングでは、稜線シワの1本による陰影の表現と2本による表現に対して、陰影付けを行った。2本の表現ではつなぎ目の部分にイラストとは違う不自然な陰影ができてしまった。距離マップを使用したレンダリングでは、輪郭シワに対して陰影付けを行った。輪郭シワの特徴である片側に対してのみ陰影付けを行うことができた。

# A b s t r a c t

|           |   |
|-----------|---|
| Title     | Real-time rendering of shadow shapes peculiar to character illustrations for clothes wrinkles |
| Author    | Ayuka Aoki  |
| Advisor   | Taichi Watanabe   |
| Key Words | illustration, rendering, shadow, shader, wrinkle  |

## [summary]

In recent years, the number of people who draw illustrations using a computer has increased, and it has become possible to finish high-quality illustrations by improving the functions of tools and developing drawing techniques. There are several important points that increase the quality of the illustration, but one of those points is shaded. The shading work requires familiarity, and even if you are accustomed to the work, there is a problem that the working time and load are large. In shading a character, shading is performed on various parts, and clothes are the most important among them. In addition, wrinkles are significant in shading clothes. In illustrations, shading that does not obey the laws of physics may be applied, and the shape of the shadow differs depending on the illustrator. In this study, we propose a method for generating illustration-specific shadows that can be applied to clothes wrinkles in order to reduce the work time and load of shading the illustrations.

Shading depends on the meaning of wrinkles. In this study, wrinkle lines are classified into two types, "edge wrinkles" and "contour wrinkles."

The shading is generated by changing the algorithm in the shadow calculation for each of the wrinkle line, "edge wrinkle" and "contour wrinkle". First, input wrinkle information. From there, create a normal map and a distance map. In the shadow calculation, the edge wrinkles are calculated using the normal map, and the contour wrinkles are calculated using the distance map. Finally, the shading is generated by combining the shading information of the calculation result.

By implementing these methods, illustration-specific shading for clothes wrinkles was realized. In rendering using the normal map, shading was applied to the expression of one wrinkle and the expression of two wrinkles. In the two expressions, an unnatural shadow different from the illustration was created at the joint. In rendering using a distance map, contour wrinkles were shaded. Shading could be performed only on one side, which is a characteristic of contour wrinkles.

# 目次

|       |                |    |
|-------|----------------|----|
| 第1章   | はじめに           | 1  |
| 1.1   | 研究背景と目的        | 2  |
| 1.2   | 論文構成           | 7  |
| 第2章   | 洋服シワと陰影について    | 8  |
| 2.1   | イラストにおける洋服シワ   | 9  |
| 2.2   | 洋服シワと陰影の形状     | 11 |
| 2.2.1 | 稜線シワ           | 11 |
| 2.2.2 | 輪郭シワ           | 14 |
| 第3章   | 提案手法           | 16 |
| 3.1   | シワの情報          | 17 |
| 3.2   | マップの作成         | 17 |
| 3.2.1 | 曲線上近傍点抽出       | 17 |
| 3.2.2 | 法線マップ          | 18 |
| 3.2.3 | 距離マップ          | 22 |
| 3.3   | 陰影計算           | 23 |
| 3.3.1 | 法線マップによる陰影計算   | 23 |
| 3.3.2 | 距離マップによる陰影計算   | 24 |
| 3.3.3 | 2種類の陰影計算の組み合わせ | 24 |
| 第4章   | 結果と考察          | 26 |
| 4.1   | 稜線シワ           | 27 |
| 4.1.1 | 1本の稜線シワ        | 27 |
| 4.1.2 | 2本の稜線シワ        | 29 |
| 4.2   | 輪郭シワ           | 31 |

|     |      |    |
|-----|------|----|
| 第5章 | まとめ  | 35 |
|     | 謝辞   | 37 |
|     | 参考文献 | 39 |

# 目次

|      |                           |    |
|------|---------------------------|----|
| 1.1  | キャラクターイラスト                | 3  |
| 1.2  | 洋服のイラスト                   | 4  |
| 1.3  | 物理法則に従わない陰影               | 5  |
| 1.4  | 光源位置のイラスト (左:光源左上, 右:光源右) | 6  |
| 2.1  | 洋服シワのイラスト                 | 9  |
| 2.2  | 陰を付けたイラスト                 | 9  |
| 2.3  | 模式図                       | 11 |
| 2.4  | 稜線シワの特徴 1                 | 12 |
| 2.5  | 断面図 1                     | 12 |
| 2.6  | 稜線シワの特徴 2                 | 13 |
| 2.7  | 断面図 2                     | 13 |
| 2.8  | 稜線シワの特徴 3                 | 14 |
| 2.9  | 断面図 3                     | 14 |
| 2.10 | 輪郭シワの特徴                   | 15 |
| 3.1  | 手法                        | 17 |
| 3.2  | 有理ベジエ曲線による稜線シワ形状          | 19 |
| 3.3  | $d$ と $\alpha$ の関係        | 20 |
| 3.4  | 曲線の微分ベクトルと法線ベクトル          | 20 |
| 3.5  | 入力データと生成した法線マップ           | 21 |
| 3.6  | 法線ベクトル                    | 22 |
| 3.7  | 入力データと生成した距離マップ           | 23 |
| 3.8  | レンダリング例                   | 24 |
| 3.9  | フローチャート                   | 25 |
| 4.1  | 1本の稜線シワ                   | 27 |

|      |                                     |    |
|------|-------------------------------------|----|
| 4.2  | 1本の稜線シワの法線マップの生成 . . . . .          | 28 |
| 4.3  | 1本の稜線シワの法線マップによるレンダリング結果 . . . . .  | 28 |
| 4.4  | 2本の稜線シワ . . . . .                   | 29 |
| 4.5  | 2本の稜線シワの法線マップの生成 . . . . .          | 30 |
| 4.6  | 2本の稜線シワの法線マップによるレンダリング結果 . . . . .  | 30 |
| 4.7  | 輪郭シワ . . . . .                      | 31 |
| 4.8  | 距離マップの生成 . . . . .                  | 32 |
| 4.9  | 距離マップによるレンダリング結果 . . . . .          | 32 |
| 4.10 | ひし形の輪郭シワ . . . . .                  | 33 |
| 4.11 | ひし形の輪郭シワの距離マップ生成 . . . . .          | 33 |
| 4.12 | ひし形の輪郭シワの距離マップによるレンダリング結果 . . . . . | 34 |

# 第 1 章

## はじめに

## 1.1 研究背景と目的

近年，コンピュータを用いてイラストを描く人が増加しており，ツールの機能向上と描画手法の発展からクオリティの高いイラストを仕上げることができるようになった．イラストを描くには様々な工程があり，完成するまでに時間が長く掛かるイラストも多い．そのため，イラスト制作に関する研究は多くあり，便利なツールの開発や，作業の自動化に関する研究が多くある．

ツールには，イラスト制作の一部の工程を自動化するものがある．ラフ画を線画にする研究 [1][2] や，線画に対して自動で着色するサービス [3] がある．また，手描きイラストの続きを描いてくれるサービス [4][5] もある．これによって，イラスト制作者は作業の一部を短縮することができる．

他にイラスト自体を自動で作成する研究がある．

実写画像からイラストを作成する研究として，写真などの画像をイラスト風にする研究 [6] や，人の顔画像から似顔絵を生成する研究 [7][8][9] がある．実写の画像をイラスト風にすることで，大量のイラストを描く作業を軽減することができる．また，用意したパーツを組み合わせることで，似顔絵イラストを生成する手法 [10] もある．

画像を用意することが難しいイラストの場合は，3D モデルを使用してイラストを作成する手法がある．3D モデルを，イラスト風の画像を生成するレンダリング [11][12][13] をすることによって，イラストを作成することができる．イラストには様々な画風があるため，画風によってレンダリングのアルゴリズムは変える必要があり，様々なレンダリングの手法が研究されている．近年では 3D モデルを使用することで，イラストを簡単に動かすこともできるようになった．

イラスト自体の生成ではイラスト制作者のタッチを表現することが難しく，様々なアルゴリズムが研究されている．

このようにイラストに関する研究は様々なものがある．

イラストには様々なジャンルがあるが、イラスト投稿サイト [14][15][16] などによく見るジャンルのイラストとしてキャラクターがある。キャラクターの中でも人間のイラストが多く投稿されている。そこで本研究では、キャラクターイラスト制作の作業の1つである陰影付けに注目した。陰影はイラストのクオリティを上げる重要なポイントの1つである。陰影付けの作業には慣れが必要であり、仮に作業に慣れたとしても作業時間や負荷が大きいという問題がある。また、陰影の形状やイラスト全体のバランスによって描き直しを必要とする反復作業も多い。

図 1.1 に 2 つのイラストを示す。左が陰影色が 1 つ、右が陰影色が 2 つ使われている。陰影色が増えることでイラストの印象は全く違い、このようにイラストにおいて陰影は重要である。



図 1.1 キャラクターイラスト

グラフィックソフトの ClipStudioPaint[17] や PhotoShop[18] には陰影色の自動調整機能や、デッサン人形を表示する補助機能があるが、陰影付けを行う機能はない。そこで、陰影付けを自

動で行うことができれば、作業時間や負荷を減らすことが期待できる。

キャラクターの陰影付けでは、様々なパーツに対して陰影付けを行うが、その中でも洋服はキャラクターの大部分を占めている。洋服の陰影付けでは、シワによって陰影の付く場所が大きく変わる。図 1.2 は洋服のイラストである。洋服に対してシワを示す線が描かれているが、その線に対して陰影がついている。このように、洋服に対して陰影を付ける場合には、シワを考慮して陰影付けを行う必要がある。

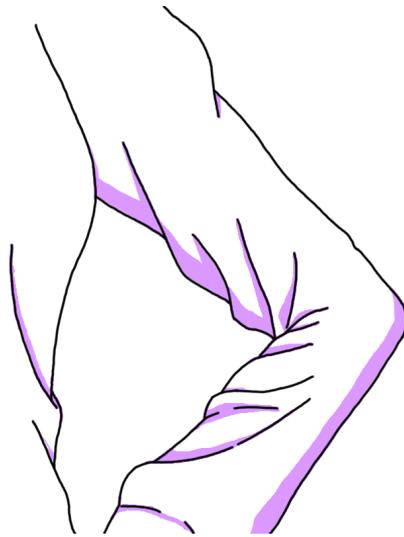


図 1.2 洋服のイラスト

またイラストでは、物理法則に従わない陰影の付け方 [19] をすることがある。図 1.3 に例を示す。イラストのように、実際には薄い影でもイラストでは立体感を表すために、はっきりと陰影を書き入れることがある。また、シワの形状をわかりやすくするために、実際には陰影はできない部分にも陰影を書き入れることがある。さらにシワの形状はイラスト制作者によって異なる。イラストに陰影を付ける時にはこうしたことも考慮する必要がある。

イラストに対して陰影を生成する研究に、Sýkora らの研究 [20] がある。この研究では線画といくつかの入力から、レリーフ彫刻のようなモデルを作成している。そしてそのモデルを使用して、陰影の計算を行う。しかしこの手法では輪郭を考慮しているため、内部にある洋服のシワに対し

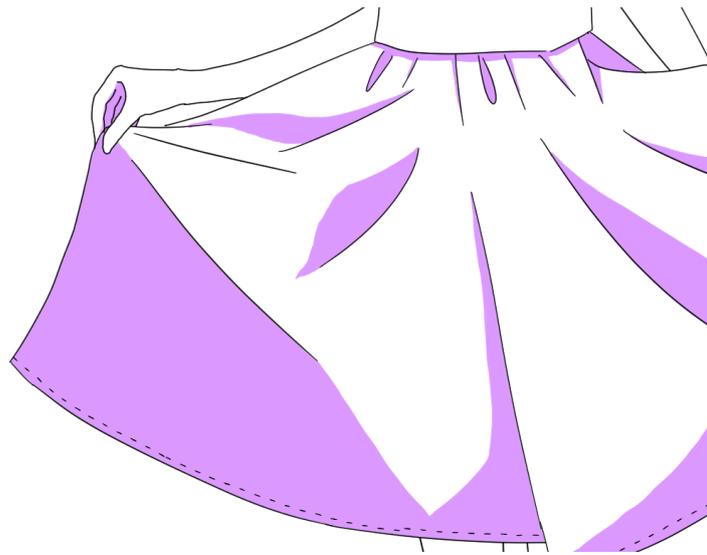


図 1.3 物理法則に従わない陰影

て陰影をつけるのは難しい。

同じようにモデルを生成するスケッチベースのモデリング手法 [21][22][23] があるが、3D モデルを生成することを目的としているため、シェーディングを行い、陰影を生成するのは難しい。

他に、Jayaraman ら [24] のイラストの内部にある洋服のシワに注目した研究がある。この研究では、ユーザーが描いたシワがどのような形状を示しているのかを分析し、そこからイラスト風のシェーディングができるモデルを生成している。しかし、アニメ塗りやそれをベースとした塗り方は、このモデルを使用して陰影を生成するには難しい場合がある。また、この研究では陰影の生成は通常のトゥーンレンダリングの手法を使用しているため、本研究の目的であるイラスト特有の陰影や制作者の好みを反映した陰影を生成するには、この手法は向いていない。

そのため、現状ではイラスト特有の陰影を生成することや、イラスト制作者が望むような陰影を生成することが難しい。

そこで本研究では、イラストへの陰影付けの作業時間や負荷の軽減を目的とし、キャラクターイラストの洋服のシワに対して、イラスト特有の陰影を自動で生成することを目指す。イラストでは様々な種類の塗り方があるが、本研究では図 1.1 のようなアニメ塗りやそれをベースとした

塗り方を対象とする。イラストでは三次元形状を必要とするような現実のような厳密な陰影ではなく、イラスト制作者の特色を反映できる陰影を望むこともある。先行研究のような正確な空間情報や 3D モデルから計算することはできない。

本研究では、後述するシワの特性を制作者が自由に設定できることによって、制作者が望む陰影の生成を可能とする。また、線画の推定では、Malik[25]の研究やモデルを生成する研究などで幅広く行われている。洋服シワに関しては Jayaraman らの研究でも行われている。そのため、本研究ではシワの分類ができているとし、陰影の計算方法について論ずる。

図 1.4 は、左図では光源が左上にある場合の陰影を付けたイラスト、右図では光源が右にある場合の陰影を付けたイラストである。光源の位置が違くと陰影の付く位置が変わるため、本研究では光源の位置も考慮して陰影の生成を行う。

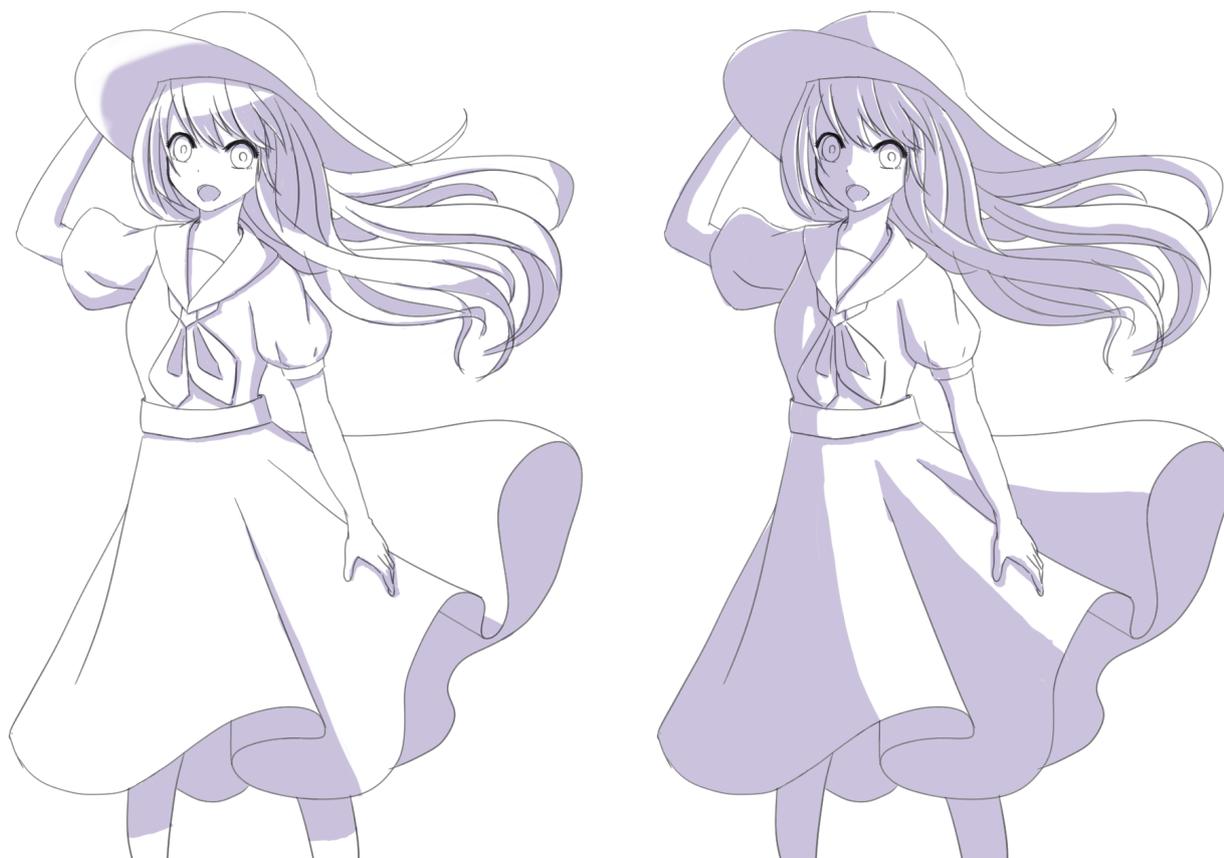


図 1.4 光源位置のイラスト (左:光源左上, 右:光源右)

## 1.2 論文構成

本論文は全 5 章で構成する。2 章で洋服シワと陰影について詳しく述べる。3 章では、提案手法について述べる。4 章では結果と考察を述べる。5 章で本研究の成果と意義をまとめ、今後の展望について述べる。

## 第 2 章

# 洋服シワと陰影について

## 2.1 イラストにおける洋服シワ

イラストにおいてシワは、見栄え良く見えるように記号化した線で表現している。図 2.1 はシワを描いたイラストである。図 2.2 は図 2.1 に陰影を付けたものである。紫色の部分は光源が上の時に、陰影が付く部分を示している。

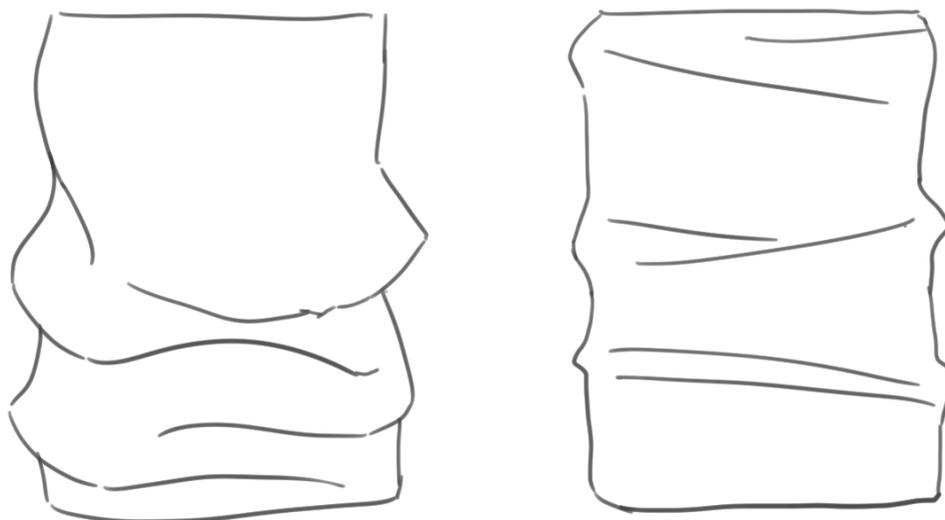


図 2.1 洋服シワのイラスト

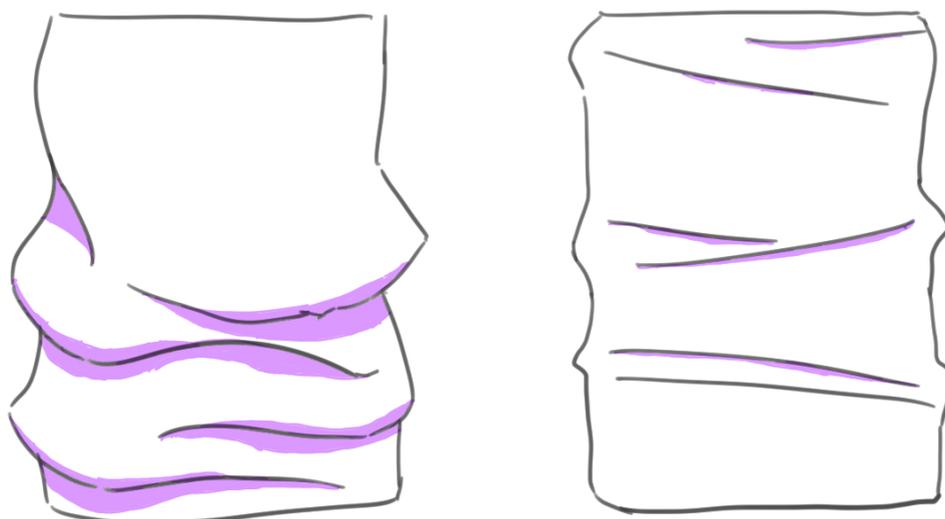


図 2.2 陰を付けたイラスト

このように陰影の付き方が違う理由は、シワが示している意味が違うためである。

図 2.3 は、下が布を上から見たときの模式図である。上が布を横から見たときの模式図である。赤の部分がシワの線が描かれている部分を示し、緑や青の部分が布の部分である。イラストのシワ線は 4 つのものを表している。布の折れ目を表しているシワ線、流れを示しているシワ線、布と空間に描かれている布の輪郭線、布と布の間に描かれている布の輪郭線の 4 つで表せる。

本研究では、布の折れ目を表しているシワ線、流れを示しているシワ線、布と布の間に描かれている布の輪郭線の、3 つに対する陰影付けを行う。これらの 3 つのシワ線のでき方を、大きく 2 種類に分けた。

図 2.3 で、実際にイラストの線画で線が書かれている部分は、赤色の部分となる。左の図では見える部分のシワ線と布が連続に続いているが、右の図では見える部分のシワ線と布が連続になっていない。このようにシワ線が左の図のような連続しているシワと、右の図のような連続していないシワの 2 つに分類する。連続しているシワを「稜線シワ」、連続していないシワを「輪郭シワ」と本研究では呼称する。

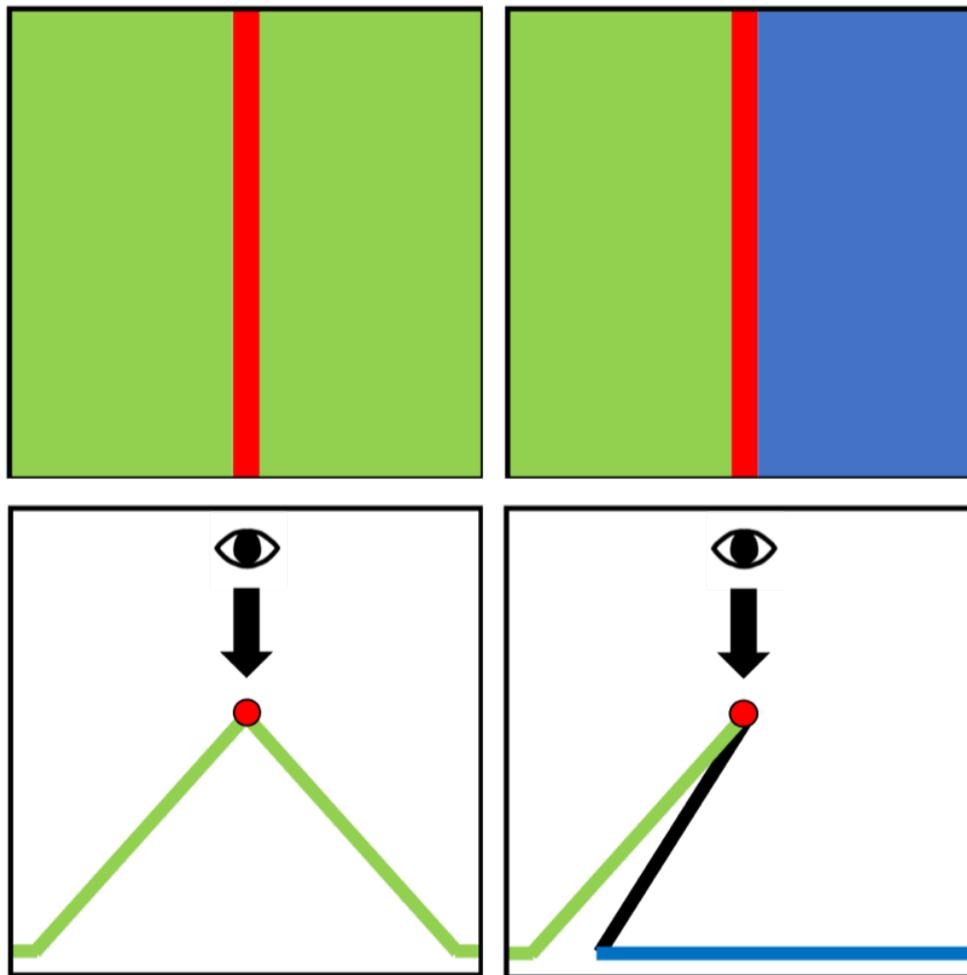


図 2.3 模式図

## 2.2 洋服シワと陰影の形状

2.1 節では稜線シワと輪郭シワに洋服のシワ線を分類した。本節では、稜線シワと輪郭シワが、イラストではどのようなシワ線の形状をし、どのような陰影を付けるのかについて、いくつかの特徴を示す。

### 2.2.1 稜線シワ

図 2.4 にシワの形状と陰影の形状の例を示す。稜線シワでは、図 2.4 の左のような形状のシワがある。これは布の山の頂点の部分のシワ線を示している。左に光源がある場合は図 2.4 の右のイ

ラストのように大きく陰影がつく特徴がある。断面図を図 2.5 に示す。赤がシワ線で、緑が布である。

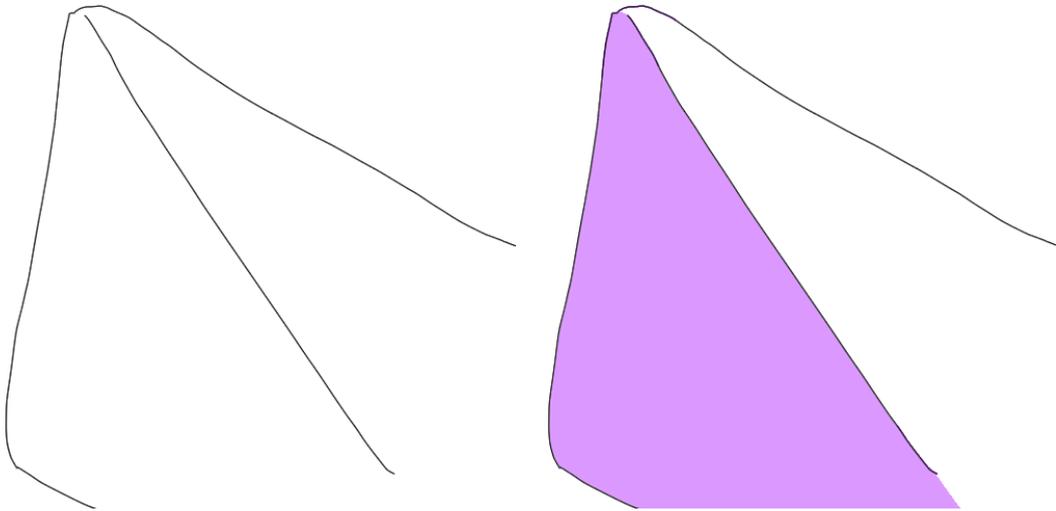


図 2.4 稜線シワの特徴 1

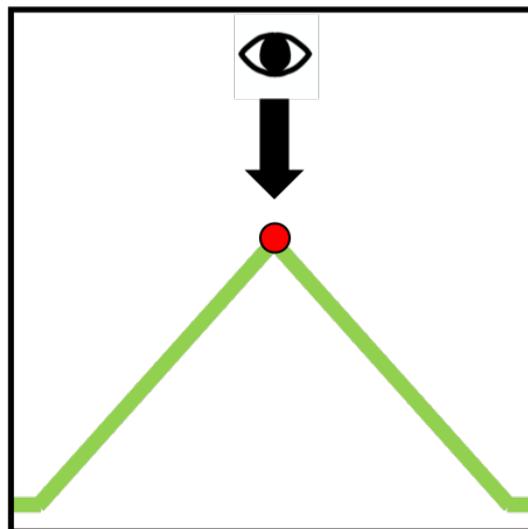


図 2.5 断面図 1

次に 2 本の稜線シワによって表される形状のシワを示す。

図 2.6 は、2 本の稜線シワによって表されるシワの形状と陰影の形状の例である。2 本の稜線シワによってできるシワ線には、図 2.6 の左のような形状のシワがある。これは 2 本のシワ線の間が山型の円弧のような形状を意味している。左に光源がある場合は図 2.6 の右のイラストのよう

に陰影がつく特徴がある。断面図を図 2.7 に示す。赤がシワ線で、緑が布である。

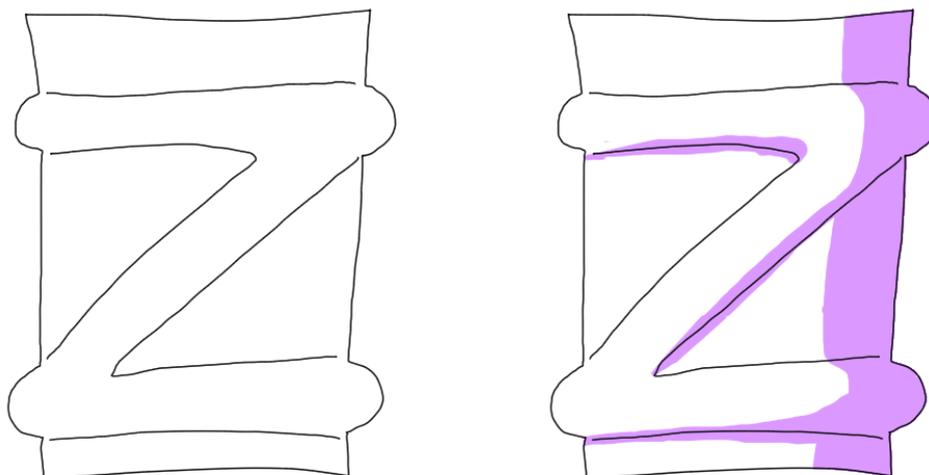


図 2.6 稜線シワの特徴 2

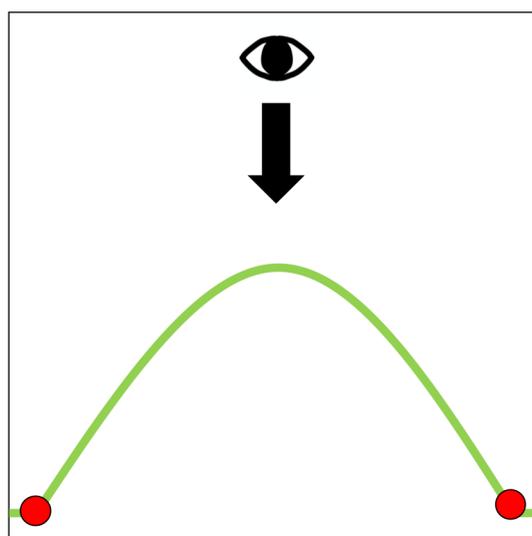


図 2.7 断面図 2

次に、図 2.8 は 2 本の稜線シワによって表されるシワの形状と陰影の形状の例の 2 つ目である。2 本の稜線シワによってできるシワ線には、図 2.8 の左のような形状のシワがある。これは 2 本のシワ線の間が谷型の円弧のような形状を意味している。左に光源がある場合は図 2.8 の右のイラストのように陰影がつく特徴がある。断面図を図 2.9 に示す。赤がシワ線で、緑が布である。

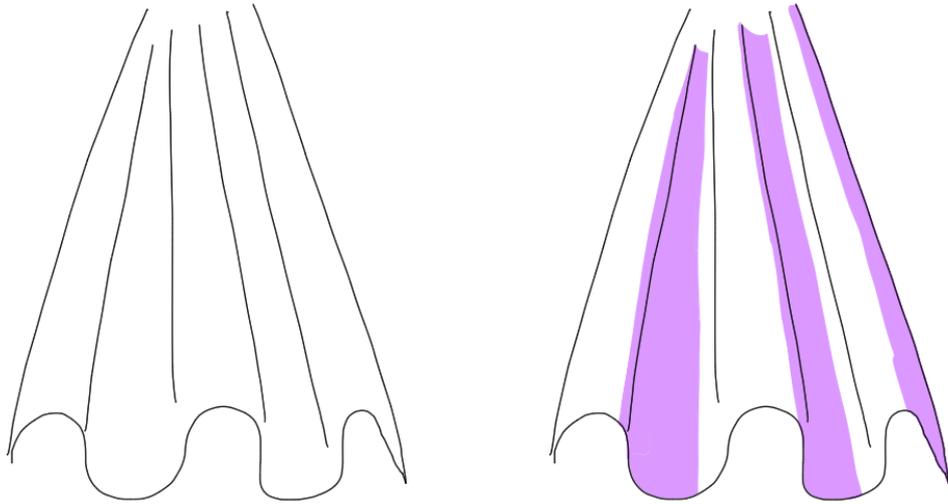


図 2.8 稜線シワの特徴 3

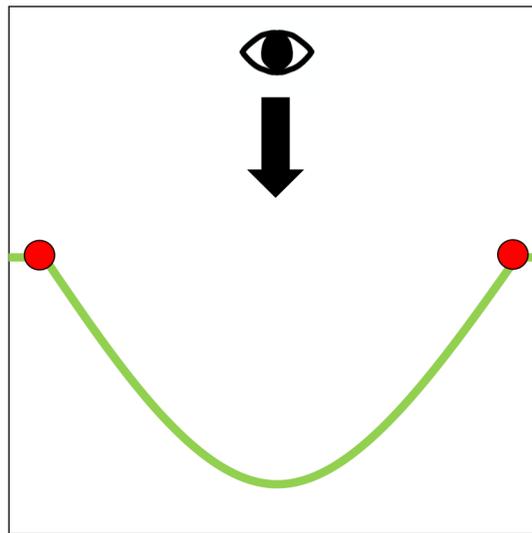


図 2.9 断面図 3

## 2.2.2 輪郭シワ

輪郭シワでは、線の片側に影が付くという特徴がある。線の両側に影が付くことはあまりない。図 2.8 の左のイラストのように同じ幅で陰影が付く場合や、図 2.8 の中央のイラストのように真ん中の部分が幅広く影が付く場合もある。また、2 つの輪郭シワが組み合わさると図 2.8 の左のイラストのような影ができるという特徴がある。

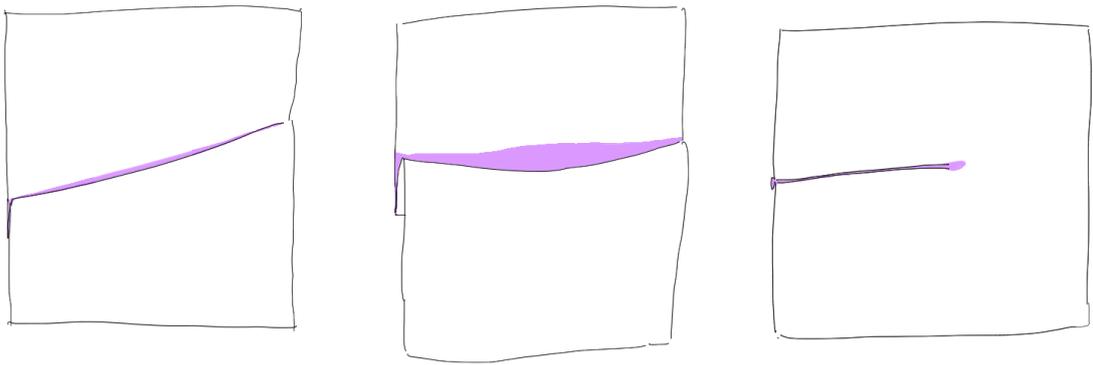


図 2.10 輪郭シワの特徴

# 第 3 章

## 提案手法

提案手法は、稜線シワか輪郭シワによって、陰影計算でのアルゴリズムを変更することで、陰影を生成する。まずシワ情報を入力する。そこから法線マップと距離マップを作成する。そして陰影計算では、作成したマップを使用してシェーダーで陰影計算を行う。稜線シワは法線マップを使用し、輪郭シワは距離マップを使用して計算を行う。最後に計算結果の陰影の情報を組み合わせて陰影を生成する。手法の流れを図 3.1 に示す。

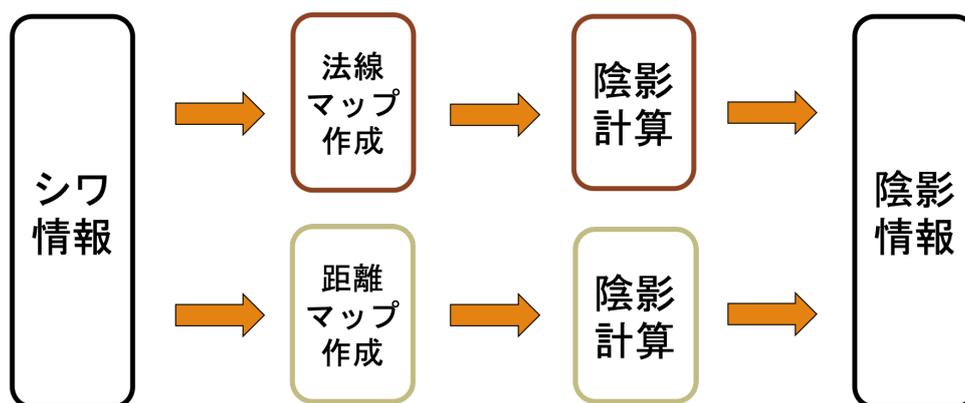


図 3.1 手法

## 3.1 シワの情報

シワの情報は任意の制御点をデータとして持つ曲線データである。本研究では、イラスト作成者が直感的に入力しやすいように、Catmull-Rom Spline 曲線 [26] のアルゴリズムを使用する。

## 3.2 マップの作成

### 3.2.1 曲線上近傍点抽出

法線マップや距離マップの作成には、任意点から曲線に対する最近傍点を求める必要がある。法線マップでは複数のシワの曲線すべてに対しての最近傍点を求め、距離マップではシワの曲線ごとに最近傍点を求める。スプライン曲線に対する最近傍点を抽出する方法は、様々な手法が提案されているが、本研究では高い精度は必要としないことや、マップ生成時における処理速度は

それほど高速でなくとも問題がないことから、適当な分割数によるサンプリング探索によって最近傍点を抽出した。以降、マップ値を算出する対象のピクセルを  $p$ ,  $p$  に対応した曲線上の最近傍点を  $C_p$ , それぞれの位置ベクトルを  $\mathbf{P}, \mathbf{C}_p$  と表記する。

### 3.2.2 法線マップ

まず稜線シワのアルゴリズムで使用する法線マップを作成する。法線マップは入力されたシワ情報を元に作成する。法線マップとは面の法線ベクトルを保存したテクスチャ画像のことである。

まず  $p$  に対する  $C_p$  を求めた後、両点間の距離  $d = |\mathbf{P} - \mathbf{C}_p|$  と単位ベクトル  $\mathbf{J} = \frac{\mathbf{P} - \mathbf{C}_p}{d}$  を求める。次に、有理ベジエ曲線 [27] によって入力したシワ線上にできる形状を設定する。有理ベジエ曲線の制御点を  $\mathbf{Q}_i$ , 各制御点の重みを  $w_i$  としたとき、有理ベジエ曲線  $\mathbf{R}(t)$  は式 (3.1) によって定まる。ここで  $B_i^n(t)$  は Bernstein 基底関数である。

$$\mathbf{R}(t) = \frac{\sum_i^n B_i^n(t) w_i \mathbf{Q}_i}{\sum_i^n B_i^n(t) w_i} \quad (3.1)$$

本手法では曲線次数を 2 とし、実数  $a, b, w$  により各制御点の位置と重み値を以下のように定めるものとする。

$$\begin{aligned} \mathbf{Q}_0 &= (-a, 0), & \mathbf{Q}_1 &= (0, b), & \mathbf{Q}_2 &= (a, 0), \\ w_0 &= 1, & w_1 &= w, & w_2 &= 1 \end{aligned} \quad (3.2)$$

これにより、稜線シワ形状は幅、高さ、曲がり具合をそれぞれ  $a, b, w$  の値を変更することにより制御することができる。図 3.2 に稜線シワ形状の模式図を示す。

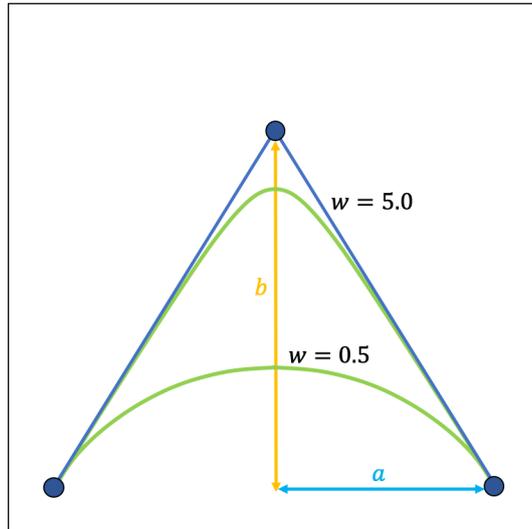


図 3.2 有理ベジエ曲線による稜線シワ形状

稜線シワは，この曲線形状を先述の Catmull-Rom Spline 曲線に沿ってスウィープしたものと定義する。

次に，稜線シワ上での法線ベクトルの求め方について述べる．法線ベクトルを求めるには，まずピクセル  $p$  の位置における有理ベジエ曲線上のパラメータ  $t$  を求める必要がある。

$\mathbf{R}(t)$  の  $x$  成分に式 (3.2) の値を代入しまとめると，以下の式を得ることができる。

$$R_x(t) = \frac{a(2t - 1)}{(t^2 - t)(2 - 2w) + 1} \quad (3.3)$$

この値が  $d$  と等しくなる場合の  $t$  を求めればよい．(3.3) を  $d$  とおいて式をまとめると，以下のような  $t$  についての 2 次方程式を得ることができる。

$$(2d - 2dw)t^2 + (2dw - 2d - 2a)t + d + a = 0 \quad (3.4)$$

この式の実数解のうち， $t$  の値が  $\frac{1}{2}$  から 1 の間にあるものが，求めるパラメータ  $t$  となる．この解を  $\alpha$  とする． $d$  と  $\alpha$  の関係の模式図を図 3.3 に示す。

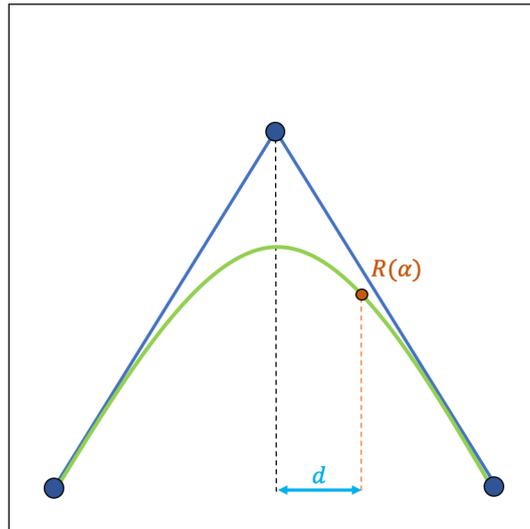


図 3.3  $d$  と  $\alpha$  の関係

次に、 $\mathbf{R}(t)$  の 1 階微分  $\mathbf{R}'(t)$  より微分ベクトル  $\mathbf{D} = \mathbf{R}'(\alpha)$  を求めておく。これを反時計回りに 90 度回転したベクトル  $(-D_y, D_x)$  が 2 次元平面上の  $\mathbf{R}(\alpha)$  に対する法線ベクトルとなる。 $\mathbf{D}$  に関する模式図を図 3.4 に示す。

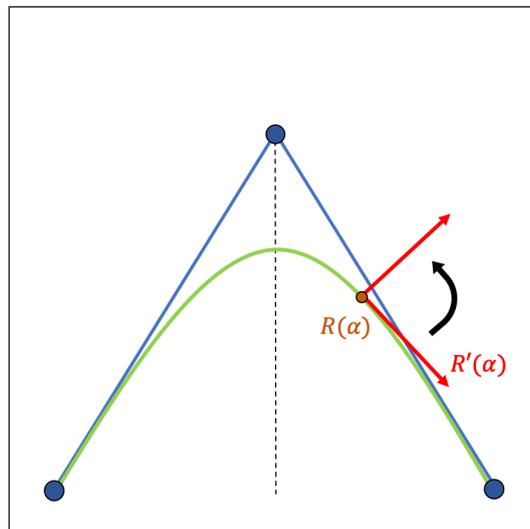


図 3.4 曲線の微分ベクトルと法線ベクトル

この  $\mathbf{D}$  と  $\mathbf{J}$  を用いて、以下のようにして法線ベクトル  $\mathbf{N}$  を求めることができる。

$$\mathbf{N}' = (-J_x D_y, -J_y D_y, D_x), \quad \mathbf{N} = \frac{\mathbf{N}'}{|\mathbf{N}'|} \quad (3.5)$$

次に，法線ベクトル  $\mathbf{N}$  をテクスチャデータに格納するための色値構造に変換する．色値  $R, G, B$  を以下のような式で求め，各成分の  $[-1, 1]$  の領域を  $[0, 255]$  に線形写像した．

$$\begin{aligned} R &= \frac{255}{2}(N_x + 1), \\ G &= \frac{255}{2}(N_y + 1), \\ B &= \frac{255}{2}(N_z + 1) \end{aligned} \quad (3.6)$$

図 3.5 は法線マップ作成の例である．シワ情報の部分が一番高い山型または一番低い谷型になるように，法線マップを作成する．図 3.5 は左がシワの情報を示した曲線であり，右が作成した山型の法線マップの例である．図 3.6 は山型の場合の法線ベクトルと RGB を示したものである．

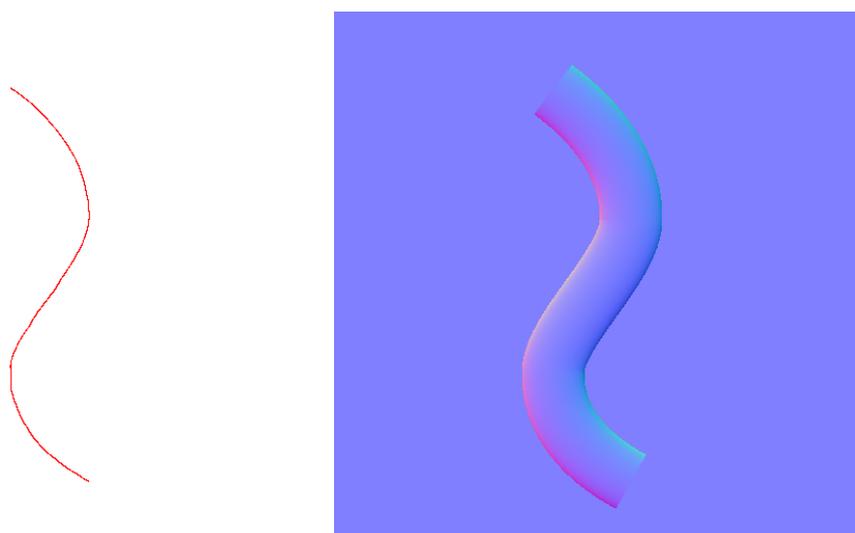


図 3.5 入力データと生成した法線マップ

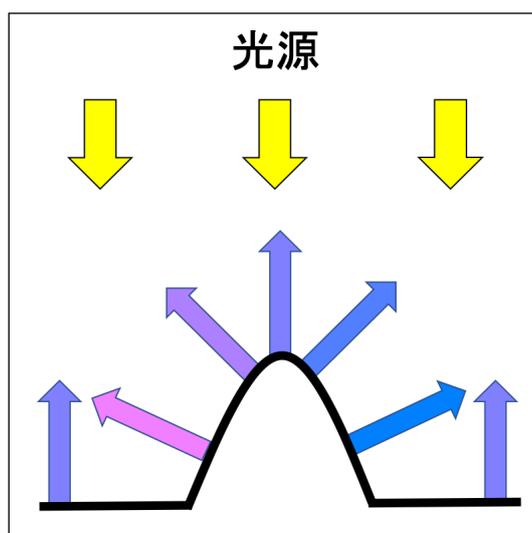


図 3.6 法線ベクトル

### 3.2.3 距離マップ

次に輪郭シワのアルゴリズムで使用する距離マップを作成する。距離マップとは、入力されたシワ曲線からの距離を保存したテクスチャ画像である。

まず  $p$  に対する  $C_p$  を求めた後、両点間の距離  $d = |\mathbf{P} - \mathbf{C}_p|$  を求める。次に曲線のどちら側に位置しているのかを、計算する。単位ベクトル  $\mathbf{J} = \frac{\mathbf{P} - \mathbf{C}_p}{d}$  を求めることで、曲線の微分ベクトルとの外積によって左右どちら側に位置しているか計算することができる。距離マップには、曲線の片側の場合、 $d$  を対応した  $p$  の位置に保存する。

作成の例を図 3.7 に示す。この距離マップでは、RGB の数値を  $(d, d, 255)$  に設定し作成した。



図 3.7 入力データと生成した距離マップ

### 3.3 陰影計算

陰影計算では円柱モデルを使用し，シェーダーで計算を行う．

#### 3.3.1 法線マップによる陰影計算

法線マップによる計算では，ランバート反射という反射モデルを使用する．面の法線ベクトルを  $\mathbf{N}$ ，光源の方向ベクトルの単位ベクトルを  $\mathbf{L}$  とする．レンダリングではその時の明度  $\mathbf{V}$  を，面の法線ベクトルと光源の方向ベクトルを反転した単位ベクトルの内積で求める．式を (3.7) に示す．

図 3.8 に図 3.5 の入力データと法線マップによるフォンシェーディングのレンダリング結果の例を示す．赤色の線が入力データとなる．本研究ではアニメ塗りを対象としているため，陰影の出力ではトゥーンレンダリングの結果を使用する．トゥーンレンダリングでは，階調を数段階程度に落としてレンダリングを行う．

$$\mathbf{V} = -\mathbf{N} \cdot \mathbf{L} \quad (3.7)$$

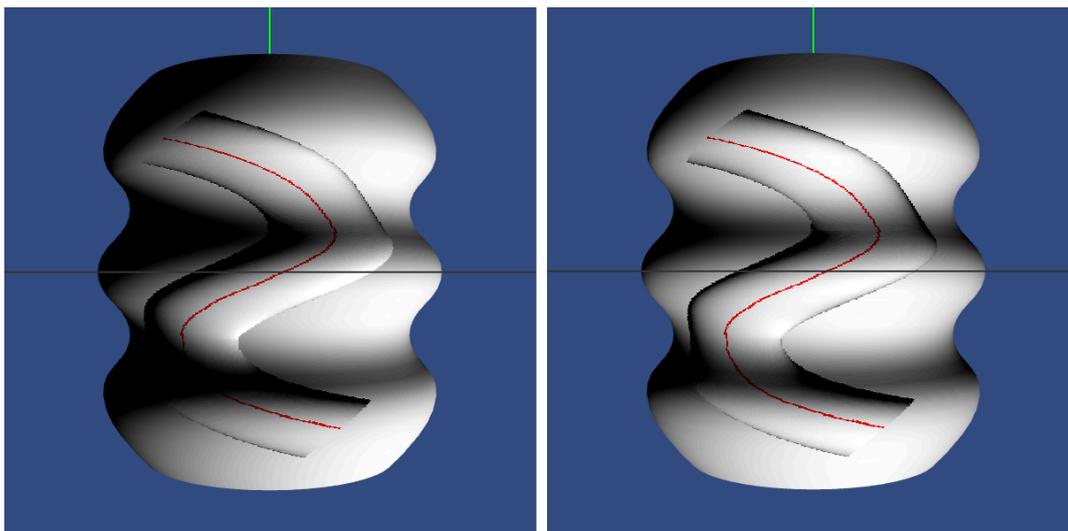


図 3.8 レンダリング例

### 3.3.2 距離マップによる陰影計算

まずシワデータに対して、シワの高さを  $h$  を設定する。そのシワの高さ  $h$  と、面の法線ベクトルと光源の方向ベクトルの角度を  $\theta$  によって、曲線からどの距離まで陰影ができるかを計算する。曲線からの距離は、距離マップから取得した数値  $d$  を使用する。その式を式 (3.8) に示す。この式 (3.8) を満たす時、陰影を表示する。

$$d \leq h \tan \theta \quad (3.8)$$

### 3.3.3 2種類の陰影計算の組み合わせ

複数のシワ線に対して陰影計算を行うとき、稜線シワと輪郭シワの両方があり、法線マップと距離マップの両方の陰影計算をする場合がある。このときの陰影計算を図 3.9 にフローチャートで示す。対象のピクセルが距離マップによって陰影が付く範囲だった場合、陰影を付ける。そうでない場合は法線マップによるトゥーンレンダリングの結果を反映する。

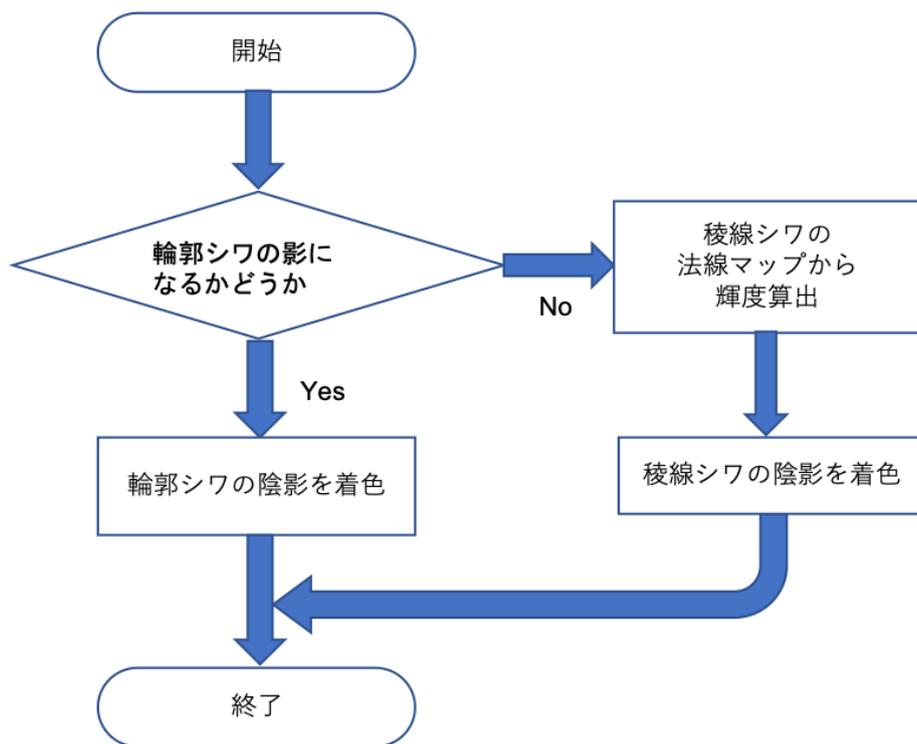


図 3.9 フローチャート

## 第 4 章

# 結果と考察

## 4.1 稜線シワ

稜線シワでは法線マップを作成し、陰影計算を行う。本研究ではアニメ塗りを対象としているため、トゥーンレンダリングを行った。明度が 0.1 以下, 0.3 以下, それ以外で色分けを行った。

### 4.1.1 1本の稜線シワ

図 4.1 は 1 本の稜線シワによって洋服シワを表現しているイラストである。これは右上に光源がある場合の陰影を紫色で塗っている。このイラストを想定し、陰影計算を行った。

図 4.2 は左が入力したシワのデータであり、右が生成した山型の法線マップである。そして図 4.3 がレンダリング結果となる。光源は図のイラストと同じように右上に設定した。各制御点の位置と重みは,  $a = 200.0, b = 500.0, w = 10.0$  で出力した。

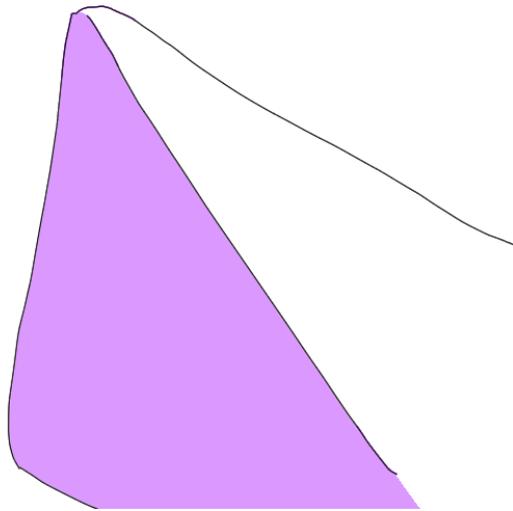


図 4.1 1本の稜線シワ

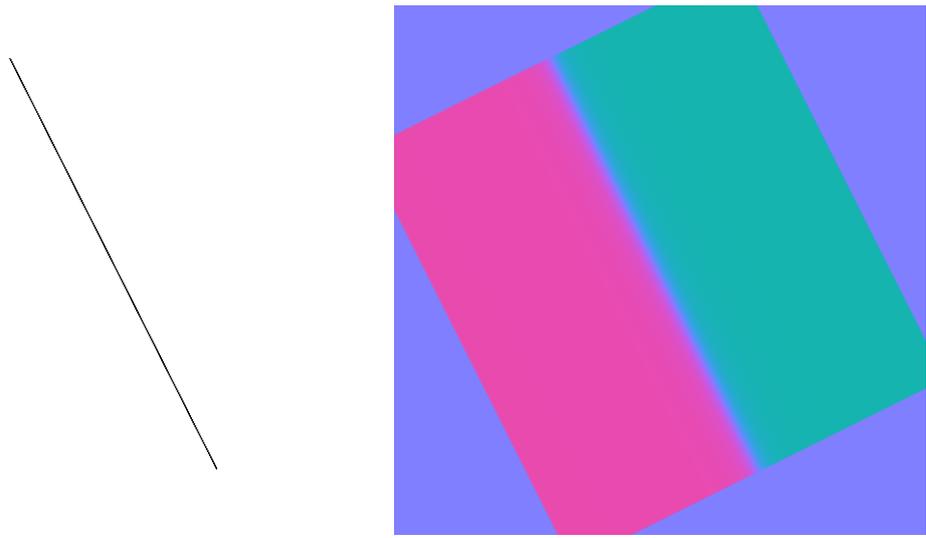


図 4.2 1本の稜線シワの法線マップの生成

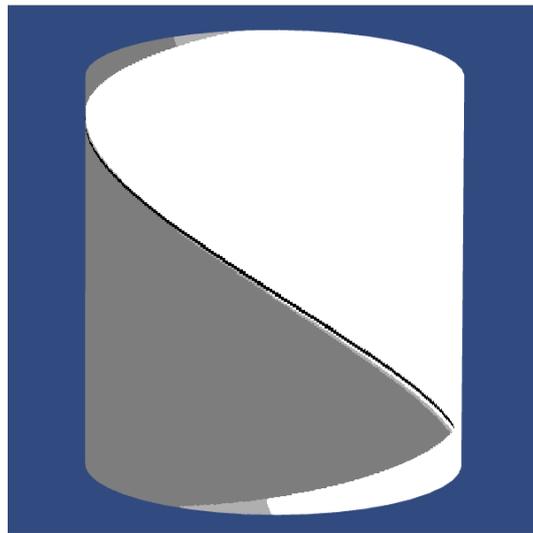


図 4.3 1本の稜線シワの法線マップによるレンダリング結果

法線マップによるレンダリングによって、稜線シワの1本のときにできる陰影を目指した。結果、シワ線の部分が高くなり、図 4.1 のイラストと同じように片側に広く陰影がついている。

これによって稜線シワの1本の場合の、布が盛り上がっている部分にできる陰影を生成できた。

### 4.1.2 2本の稜線シワ

図 4.4 は 2 本の稜線シワによって洋服シワを表現しているイラストである。これは左上に光源がある場合の陰影を紫色で塗っている。このイラストを想定し、陰影計算を行った。

図 4.5 は左が入力したシワのデータであり、右が生成した 2 本の稜線シワの法線マップである。そして図 4.6 がレンダリング結果となる。光源は図のイラストと同じように左上に設定した。各制御点の位置と重みは、 $a = 20.0, b = 100.0, w = 0.3$  で出力した。

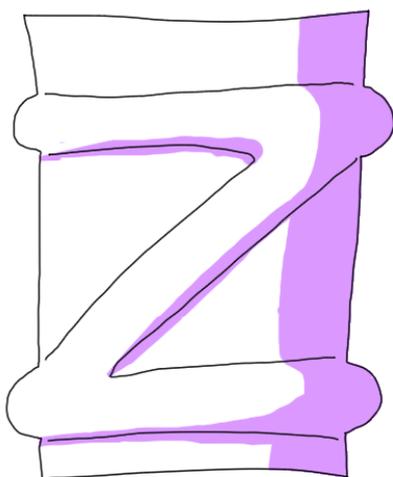


図 4.4 2本の稜線シワ

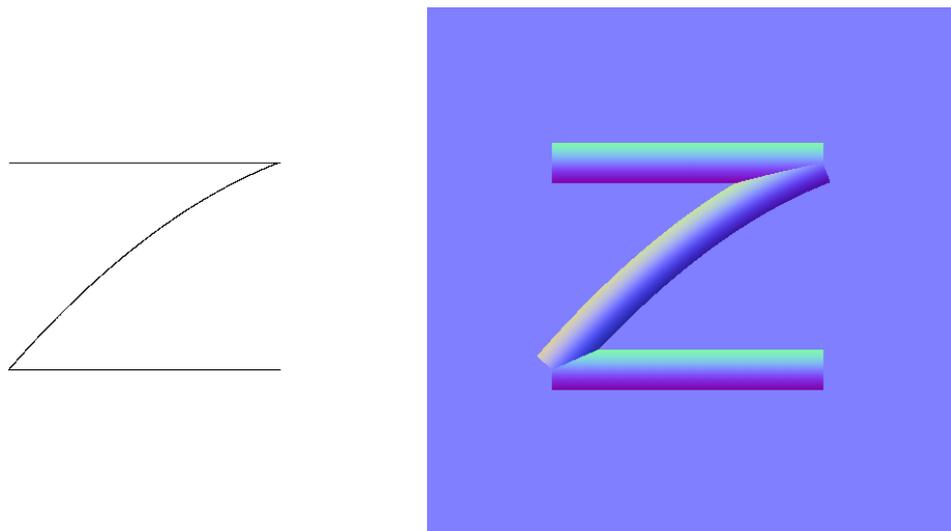


図 4.5 2本の稜線シワの法線マップの生成

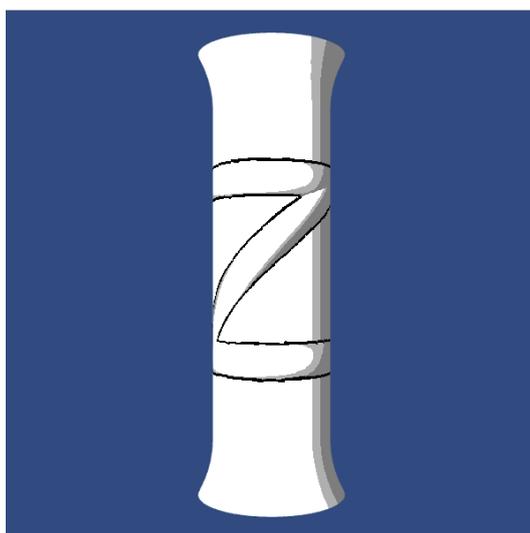


図 4.6 2本の稜線シワの法線マップによるレンダリング結果

2本の稜線シワの間が盛り上がっている図4.4のイラストのような、丸みを帯びた陰影を生成することができた。また左上の光源の影響も反映できている。これによって稜線シワの2本の場合の、布が盛り上がっている部分にできる陰影を生成できた。しかし、稜線シワのつなぎ目の部分に、イラストとは違う不自然な陰影ができてしまっている。これは、法線マップの作成でつなぎ目ができる形状を設定しているためである。

## 4.2 輪郭シワ

輪郭シワでは距離マップを作成し、それを使用して陰影計算を行う。図 4.7 は輪郭シワに対して陰影を付けたイラストである。このイラストのような輪郭シワに対して陰影付けの計算を行う。

図 4.8 は左が入力したシワのデータであり、右が生成した距離マップである。そして図 4.9 がレンダリング結果となる。図のイラストは光源は右上にあるため、光源は同じ右上に設定をし、レンダリングを行った。シワの高さ  $h$  は 0.01 に設定した。

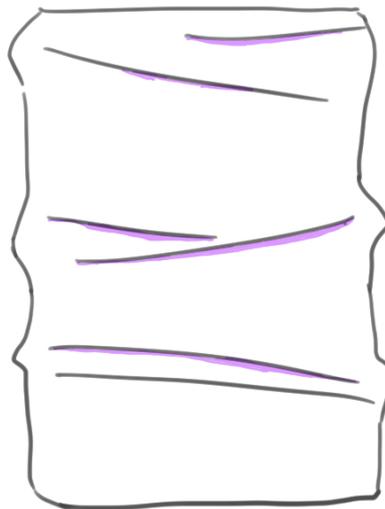


図 4.7 輪郭シワ



図 4.8 距離マップの生成

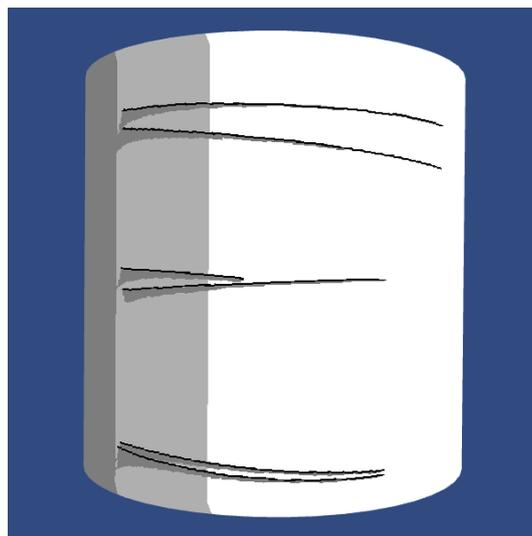


図 4.9 距離マップによるレンダリング結果

結果，図 4.7 のイラストと同じような陰影を生成することができた．輪郭シワの特徴であるシワ線の片側に対してのみ，陰影の生成ができています．

次に，2本の輪郭シワがあり，2本の間が折り込まれているような図 4.10 のようなひし形のようなシワ線のイラストも想定し，計算を行った．図 4.11 は左が入力したシワのデータであり，右が生成した距離マップである．そして図 4.12 がレンダリング結果となる．光源は同じ右に設定をし，レンダリングを行った．シワの高さ  $h$  は 0.01 に設定した．

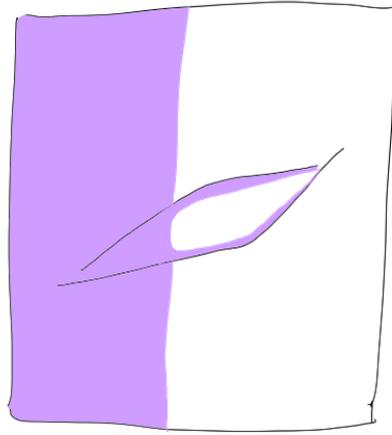


図 4.10 ひし形の輪郭シワ

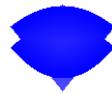


図 4.11 ひし形の輪郭シワの距離マップ生成

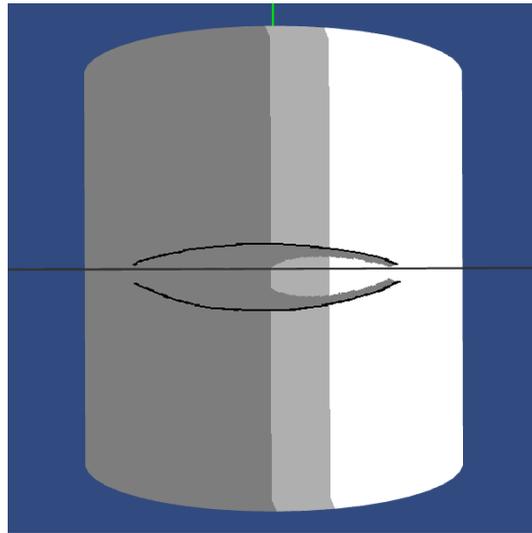


図 4.12 ひし形の輪郭シワの距離マップによるレンダリング結果

イラストのようにシワ線に隣接している部分は陰影ができ、少し離れている平らになる部分は陰影がついていない。また、シワ線の片側に対してのみ陰影の生成ができています。図 4.10 のイラストのひし形のようなシワ線に対する陰影を生成できた。

## 第 5 章

## まとめ

本論文では、キャラクターイラストの洋服のシワに対して、イラスト制作者が望む陰影を自動で生成する手法を提案した。シワを分類し、その分類したシワに対してアルゴリズムを変更することで、洋服シワに対するイラスト特有の陰影付けを実現した。また、幅、高さ、曲がり具合などをいくつかの値を変更することで制御でき、イラスト特有の陰影の生成や、イラスト制作者の特徴を反映した陰影を生成することが可能となった。

しかし、稜線シワが重なる部分では、理想の陰影をだすことが難しいという課題がある。今後は、稜線シワでの法線マップの作成の際に、稜線シワの重なる部分での処理を変更する必要があると考える。

# 謝辭

本研究を進めるにあたり、多くのアドバイスや指導をしてくださった本研究室の先生方、先輩方に心より感謝いたします。また、相談に乗ってくれた友人達にも深く感謝いたします。

## 参考文献

- [1] 黒岩直子, 藤代一成. 複数の主曲線抽出アルゴリズムを搭載した自動ペン入れシステムの開発. 情報処理学会第 67 回全国大会講演論文集, No. 1, pp. 173–174, 2005.
- [2] 小川徹, 山崎俊彦, 相澤清晴. イラスト制作支援に向けたラフ画像の線画化. 情報処理学会第 78 回全国大会講演論文集, No. 1, pp. 221–222, 2016.
- [3] Inc Preferred Networks. paintschainner. [https://paintschainer.preferred.tech/index\\_ja.html](https://paintschainer.preferred.tech/index_ja.html). 参照: 2020.01.28.
- [4] David Ha and Douglas Eck. A neural representation of sketch drawings. *ICLR*, 2017.
- [5] Google LLC. sketch-rnn. [https://magenta.tensorflow.org/assets/sketch\\_rnn\\_demo/index.html](https://magenta.tensorflow.org/assets/sketch_rnn_demo/index.html). 参照: 2020.01.28.
- [6] 濱崎雄介, 近藤邦雄. 実写画像をもとにしたイラストレーションの生成手法. 図学研究, Vol. 37, pp. 15–16, 2003.
- [7] 西郷木綿子, 佐藤美恵, 春日正男, 三輪祥太郎, 羽島一夫. 顔特徴点を利用した表情のある似顔絵生成手法. 映像情報メディア学会技術報告, Vol. 26, No. 69, pp. 13–16, 2002.
- [8] 川出雅人, 細井聖, 田畑尚弘, 秋間正道. イラストレータの画風を実現する似顔絵自動作成技術. 電子情報通信学会技術研究報告. HIP, ヒューマン情報処理 97 (117), pp. 33–40, 1997.
- [9] 山路憲, 萩原将文. 顔画像を用いた人物カラーイラスト作成システム. 電気学会論文誌. C, Vol. 122, No. 5, pp. 792–798, 2002.
- [10] 小松璃子, 伊藤貴之. パーツ単位のモーフィングによる似顔絵生成. 芸術科学会論文誌, Vol. 14, No. 5, pp. 180–187, 2015.
- [11] Philippe Decaudin. Cartoon-looking rendering of 3d-scenes. *Research Report*, 1996.
- [12] 佐藤陽悦, 藤本忠博, 村岡一信, 千葉則茂. ポリゴンベース幾何モデルの水墨画調レンダリング法. 情報処理学会研究報告. CG, グラフィクスと CAD 研究会報告, pp. 37–42, 2001.
- [13] 村上恭子, 鶴野玲治. 顔料及び支持体の特性を考慮したパステル画風レンダリング. 芸術科学

会論文誌, Vol. 1, No. 2, pp. 89–96, 2002.

- [14] pixiv Inc. pixiv. <https://www.pixiv.net/>. 参照: 2020.01.28.
- [15] TINAMI 株式会社. TINAMI. <https://www.tinami.com/>. 参照: 2020.01.28.
- [16] Pipa.jp Ltd. GALLERIA. <https://galleria.emotionflow.com/>. 参照: 2020.01.28.
- [17] CELSYS Inc. Clip studio paint. <https://www.clipstudio.net/>. 参照: 2019.02.15.
- [18] Co., Ltd Adobe Systems. adobe photoshop cc. <https://www.adobe.com/jp/products/photoshop.html>. 参照: 2019.02.15.
- [19] ダテナオト. デジタルツールで描く!服のシワと影の描き方. マイナビ出版, 東京都, 2018.
- [20] Daniel Šýkora, Ladislav Kavan, Martin Čadík, Ondřej Jamriška, Alec Jacobson, Brian Whited, Maryann Simmons, and Olga Sorkine-Hornung. Ink-and-Ray: Bas-relief meshes for adding global illumination effects to hand-drawn characters. *ACM Transaction on Graphics*, Vol. 33, No. 2, 2014.
- [21] Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. Teddy: A sketching interface for 3d freeform design. *SIGGRAPH'99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 409–416, 1999.
- [22] Andrew Nealen, Takeo Igarashi, Olga Sorkine, and Marc Alexa. Fibermesh: Designing freeform surfaces with 3d curves. *ACM Transactions on Graphics*, Vol. 26, No. 3, 2007.
- [23] Yotam Gingold, Takeo Igarashi, and Denis Zorin. Structured annotations for 2d-to-3d modeling. *ACM Transactions on Graphics*, Vol. 28, No. 5, 2009.
- [24] Pradeep Kumar Jayaraman, Chi-Wing Fu, Jianmin Zheng, Xueting Liu, and Tien-Tsin. Wong globally consistent wrinkle-aware shading of line drawings. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 24, No. 7, pp. 2103–2117, 2018.
- [25] J Malik. Interpreting line drawings of curved objects. *International Journal of Computer*

*Vision*, Vol. 1, p. 73–103, 1987.

- [26] Phillip J. Barry and Ronald N. Goldman. Recursive evaluation algorithm for a class of catmull-rom splines. *ACM SIGGRAPH Computer Graphics*, Vol. 22, No. 4, 1998.
- [27] G.E.Farin. NURBS - 射影幾何学から実務まで - 第 2 版. 共立出版, 2001.