

2D キャラクターの自動陰影付け

蔣 昌磊[†] 王 旭[‡] 柿本 正憲^{‡,*}

[†] 東京工科大学大学院 〒192-0914 東京都八王子市片倉町 1404-1

[‡] 東京工科大学 ‘ドワンゴ CG リサーチ

E-mail: [†] g3116005e6@edu.teu.ac.jp, [‡] m02171017e@edu.teu.ac.jp [‡] kakimotoms@stf.teu.ac.jp

あらまし 本研究では、TCDCN というニューラルネットワークを用いた機械学習を導入し、ユーザーが 2D キャラクターを描く際の陰影付け支援する手法を提案する。機械学習の画像認識能力を通じて、アニメキャラクターの顔特徴点の位置情報を求める。次に、獲得した位置情報を分析して、設定した光源の位置を合わせてキャラクターの陰影生成区域を確定し、自動的に陰影を付ける。本研究により手描きアニメ制作効率の向上が期待できる。

キーワード 陰影付け 機械学習 TCDCN

The Shading of 2D Characters

Changlei JIANG[†] Xu WANG[‡] and Masanori KAKIMOTO^{‡,*}

[†] Tokyo University of Technology Graduate School 1404-1 Katakura, Hachioji, Tokyo, 192-0914 Japan

[‡] Tokyo University of Technology ‘DWANGO CG Research

[†] g3116005e6@edu.teu.ac.jp, [‡] m02171017e@edu.teu.ac.jp [‡] kakimotoms@stf.teu.ac.jp

Abstract In this research, we introduce a shading support method for animators or illustrators who draw 2D characters. Our method utilizes machine learning using TCDCN neural network. Through image recognition ability of machine learning, our system obtains position information of facial feature points of an animated character. Next, the system analyzes the acquired position information and the position of the pre-defined light source, and determines the shadow or shade generation area of the character and finally applies the shading on the 2D characters. This research could be expected to improve hand-drawn animation production efficiency.

Keyword Shading Machine Learning TCDCN

1. はじめに

1.1. 研究の背景

アニメ制作の中にあるキャラクターの陰影付け作業は経験が重要で、新人イラストレーターが陰影をうまく描くのは難しいことである。そして、たとえ経験者でも、手作業による伝統的な陰影付け手法はかなり時間がかかる。陰影付けの形状とキャラクター全体のバランスを考えて、大きい範囲で描き直すという作業もしばしば生じる。

一方、近年画像処理分野でディープラーニング技術が急速に普及しつつある。ディープラーニングまたは深層学習とは、多層のニューラルネットワークによる機械学習手法である。ジェフリー・ヒントンから始まる多層ニューラルネットワークの学習の研究や、学習に必要な計算機の性能増加、そしてインターネットの発展によって、学習データの収集と学習が容易になった。そして重要なのは、その結果、音声・画像・自然言語を対象とする問題に対して、伝統な手法よりも学習による手法がはるかに高い性能を持っている、とい

うことである。このような斬新な技術を活用する研究が盛んにおこなわれつつある。

1.2. 研究の目的

本研究では、手作業で陰影付けが難しいという問題点に対して、機械学習による解決方法の適用に着目する。ディープラーニングの画像処理能力を利用して、2D キャラクター制作時の必要な陰影付け効果を自動的に生成するツールがあれば、アニメ制作のうちイラストレーターがキャラクターの絵を描く場合の効率は今より上がることが期待できる。

多数のキャラクターアニメーション作品を筆者らが分析調査した結果によれば、アニメやイラストのキャラクターの陰影付けは顔上で行う場合が多い。顔は作品中で鑑賞者が最も見る頻度が高く、微妙なニュアンスを伝えるために陰影付けを多用するためと考えられる。顔の陰影付け作業の効率化は重要な課題である。一方で、顔には、目、鼻、眉など多くのパーツがあるため、陰影付けの自動化は自明な問題ではない。そこで本研究では、機械学習を利用して、パーツ配置や顔

の向きの影響を受けにくいアニメキャラクター顔の陰影付けの自動化に取り組む。

2. 従来研究

Sun ら [1] は、独自設計した 3 層構造の CNN (Convolutional Neural Networks) を使って、人の実写顔画像の特徴マーキングを実現した。Sun らは学習データの目、鼻、口の座標情報を三つに分けて、別々に三つの CNN にそれぞれ学習させ、最後に統合する方法で人の顔特徴マーキングを実現した。

Zhang ら [2] も機械学習で人の顔特徴マーキングについての手法を発表した。補助タスクという概念と通常の CNN とを組み合わせ、TCDCN (Tasks Constrained Deep Convolutional Network) という新しいニューラルネットワークを提案した。Zhang らの手法は以前の CNN による顔特徴マーキングの結果よりも良好な結果が得られる傾向にある。特に、手や、サングラス、他の人や物体にブロックされた顔写真、もしくは側顔に対する処理能力は従来の CNN より大きく上がっている。

ultraist[3]は、2009 年に Imager::AnimeFace というアニメ顔検出ライブラリをネット上で公開した。このライブラリは Perl から使えるライブラリで、画像の中からキャラクターの顔、目、口を自動的に見つける。Imager::AnimeFace は広く一般に利用されている。

nagadomi[4] が github 上で公開した lbpcascade_animeface 法は、機械学習を用いてアニメキャラクターの顔領域を自動的に検出することができる。本研究では、学習データの顔画像領域を多数のアニメ画像から抽出する前処理において lbpcascade_animeface 法を利用している。

3. キャラクター陰影付け手法の概要

本研究で実現する顔の陰影付け手法は、アニメキャラクター画像に対して、機械学習結果を用いて、事前に設定した顔特徴点を識別する。次に識別した特徴点の xy 座標を通じてキャラクターの顔特徴線をフィッティングする。最後に陰影生成区域を確定し、自動的に陰影を付ける。

3.1. 機械学習による顔特徴点識別

顔特徴点とは図 1 のように顔上の目、口、鼻、眉、顔輪郭などの境界上の位置を代表する数個から数十個のサンプル点のことである。

提案手法では、事前手作業でアニメキャラクターの顔特徴点を指定し、それらの xy 座標を学習データとして TCDCN 与え学習させる。学習された TCDCN の識別タスクを用いて未知の顔画像に対して顔特徴をマーキングし、その未知の顔画像の顔特徴点を得る。



図 1 特徴点説明例

補助タスクは、TCDCN で顔特徴マーキングする際、識別タスクの結果をより良くするために追加されるネットワーク層である。Zhang らの実験では、人の写真に対して、男性もしくは女性の区別、眼鏡の有無、顔向きの情報などを学習した補助タスクが用いられた。提案手法では、目、口、顔向きの状態分類情報を学習データとして補助タスクに与える。識別タスクを学習すると同時に、補助タスクも同時に学習し、損失関数に貢献して、識別タスクの学習結果の精度を上げる。

3.2. キャラクターの顔特徴線フィッティング

顔の特徴線は、図 2 のように顔の構造と顔特徴器官などをもとにして、顔の向き情報、器官位置情報を推測できる中心線や眉線などの補助線である。

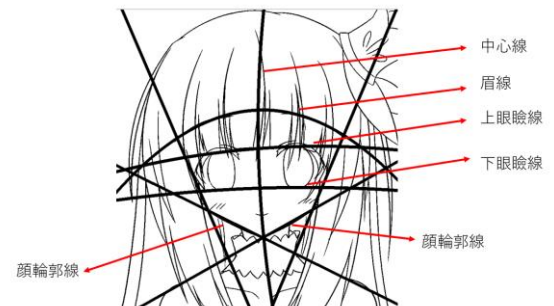


図 2 特徴線説明例

識別タスクの認識結果として得られたすべての特徴点の座標情報を使って、2 次曲線あてはめを行って、顔特徴線を算出する。

3.3. 目標区域の指定と影付け

キャラクターの顔特徴線を計算した後、陰影付けの目標区域を設定する。図 3 (A) に陰影付けの例を示す。ここでは、キャラクターの右顔を陰影付けの目標とすると想定する。特徴線の中心線、眉線、顔輪郭線の二次方程式と線形計画法を使ってキャラクターの右顔区域を確定して陰影色を付ける。このとき、中心線方程式の係数を調整すると、図 3 (B) のように陰影の形を変える。これによりデザイナーは生成する陰影位置を容易に調整できる。

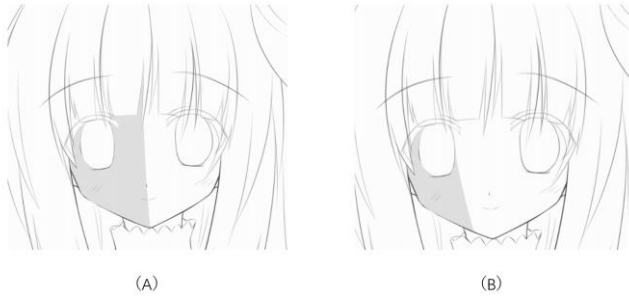


図3 (A) 線形計画法で影色を塗る
(B) 係数調整による影領域変形

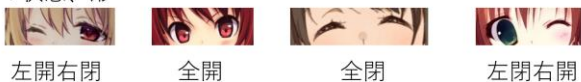
4.機械学習による顔特徴のフィッティング

3章で述べた手法のようにアニメキャラクターの顔に対してパーツ分けするためには、図1のような仕組みの顔特徴情報が必要である。本研究では、機械学習技術を利用してキャラクターの顔特徴情報を獲得する。

4.1.学習データの準備

アニメキャラクターの顔特徴と現在よく使われているニューラルネットワークの性能を分析し検討した結果、筆者は、ZhangらのTCDCNニューラルネットワークを利用すれば、アニメキャラクターに対応できると想定した。TCDCNの特徴である複数補助タスクの設置によって識別精度向上が期待できる、という点が挙げられる。アニメキャラクターの作者によって様々な作画スタイルがあるという現状で、それらのスタイルを踏まえた補助タスク設定を通じてニューラルネットワークに学習させるという手法を提案する。本研究では、キャラクターの描き方や口、目の開き状態に応じて、一つ一つの補助タスクを設定し、識別精度を上げることとする。補助タスクの学習の詳細については4.2で述べる。

目の状態、形



左開右閉 全開 全閉 左開右閉

口の状態、形



閉 開 点状

顔向き



右大きい偏り 右偏り 真ん中 左偏り 左大きい偏り

図4 補助タスクに与えたパーツ分類と状態の種類

補助タスクで使用するパーツの状態を図4に示す。キャラクターの目の状態は4種類、口の状態は3種類、そしてキャラクターの顔向きの状態は5種類である。学習データとしては、各キャラクター顔画像について目、口、顔向きの状態番号を用意する。状態番号は目視により設定する。

次に、識別タスクの学習データに使用する各パーツ特徴点について説明する。特徴点数を固定とし、通し番号による順番付けを行った。一つのキャラクターについて図5の標記順で顔16点、鼻1点、口9点、目12点、眉10点、合計45点の特徴点データセットを用いる。事前手作業により、330枚のアニメキャラクター画像に対する特徴点学習データを準備した。合計14,850点を手作業で指定したことになる。画像データは学習用とテスト用に使い分けしている。330枚のうち七割が学習用、三割がテスト用である。

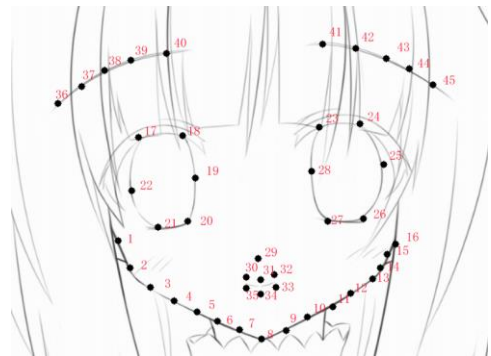


図5 学習データにおける特徴点の番号付け

4.2.特徴点座標の前処理

ニューラルネットワークに学習させるために幾つの前処理作業が必要である。

まず、キャラクターの顔部分をタイトに囲む顔画像を用意する必要がある。提案手法ではgithub上でnagadomi[4]が公開したlbpcascade_animeface法を採用し顔画像を切り出した。

次に、獲得した顔画像に対して、元画像の手作業でマーキングした特徴点の座標情報を顔画像の座標系に変換を行う。さらに、すべての特徴点の xy 座標を次式に基づき処理して平均0、分散1の標準正規分布に従う正規化処理を行う。

$$x = \frac{x_0 - \bar{x}}{\sigma} \quad (1)$$

ここで、 x_0 は手作業マーキングした特徴点の座標値、 \bar{x} は全座標平均値、 σ は座標標準偏差である。 y 座標についても同様に処理する。

正規化後、すべての特徴点の画素の座標を、事前に決定した順番に従い、先に45個の x 座標、続いて、45個の y 座標合計90個の数値を一つの数列として保存する。この数列は当該画像の特徴点列の正解データであり、以降landmarkと呼ぶことにする。一方で、すべての顔画像に対し、解像度 40×40 の画像に拡大または縮小変換して学習用データとする。

特徴点を正規化しないlandmarkで学習した場合は約10時間かかったのに対し、正規化したlandmarkで

学習する時間は約 10 分に短縮できた。

4.3.TCDCN の実装

本研究のニューラルネットワークは Zhang らの 3 層構造ニューラルネットワークを参考して、図 6 で示した構造を設計して実験を行った。全結合層に入力する前段で、コンボリューション処理により 40×40 の顔画像を 64 枚の 2×2 特徴マップに変換した。

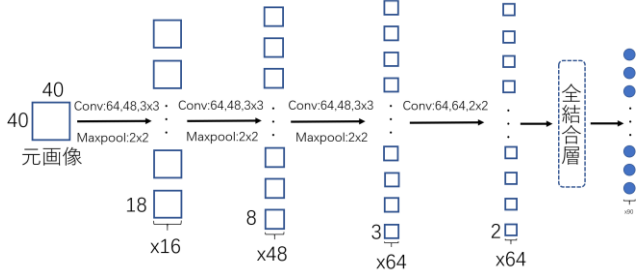


図 6 ニューラルネットワーク構造

提案手法では、毎回 Maxpooling で処理する前にすべての特徴マップのデータに対して、RELU 関数処理を施した。

2×2 の各特徴マップについて行優先順で長さ 4 の一次元数列に展開し、総計 256 個のデータを全結合層の入力層に与える。

識別タスクの全結合層は入力層から一つ目の全結合を経て 100 個のノードからなる中間層に変換する。続いて、二つ目の全結合を行って、90 個の数値の数列を出力層の結果として得る。

識別タスクの学習時に用いる損失関数は次式を採用した。

$$E_{\text{det}} = \frac{1}{2} \sum_{n=1}^{N_{\text{det}}} \|y_n - d_n\|^2 \quad (2)$$

ここで、 N_{det} は識別タスクの全結合層出力数列の大きさであり landmark 数列の大きさである、 y_n は出力値、 d_n は landmark 値である。

出力層から出力した 90 個の数値データと入力画像の landmark (学習用特徴点 xy 座標) の各数値を各自対応して式 3 を通じて識別タスクの損失関数の値を求める。

補助タスクについては識別タスクとは別個の全結合層を用意する。4.1 で述べたようにアニメキャラクターの特徴を分析し、目の向き、口の向き、顔の向き、合計三つの項目を設けた。そして、三つの項目をそれぞれ TCDCN の三つの補助タスクに与える。目を例とすると、全開、全閉、左開右閉、左閉右開という四つの状態がある。この四つの状態に対して事前に合計 4 個の数値を割り当てることとする。対応する属性が true の時、対応する値を 1 にする。false の時、対応の値

を 0 にする。口、顔向きも同じ規則で三つの補助タスクの正解数列を学習データとして与える。

三つの補助タスクそれぞれに対して、識別タスクとは別個の全結合層を用意する。補助タスクの出力層のノード数は補助タスクの正解数列の長さすなわち状態の種類数に対応する。例えば、補助タスク (目) に対して、出力層で出力するノード数は 4 である。

補助タスクの場合、状態を表現する論理値データを学習させるため、毎回の学習時の出力層データに対し、一度 softmax 関数で処理する必要がある。全結合層出力数列 y_i に対する softmax 関数 P_i は次式のようにになる。

$$P_i = \frac{e^{y_i}}{\sum_{k=1}^{N_{\text{aux}}} e^{y_k}} \quad (3)$$

ここで N_{aux} は各補助タスク用全結合出力層の数列の大きさである。例えば、目の状態を学習させる補助タスクでは $N_{\text{aux}} = 4$ である。口、顔向きの状態の補助タスクではそれぞれ $N_{\text{aux}} = 3$, $N_{\text{aux}} = 5$ である。softmax 関数の出力数列は全要素が正で合計が 1 になる特性を持つ。

こうして得た数列 P_i と入力画像の対応補助タスク正解数列 d_i から、次式を通じて補助タスクの損失関数値 E_{aux} を計算する。

$$E_{\text{aux}} = \frac{1}{N} \sum_{n=1}^N [d_i \ln P_i + (1 - d_i) \ln(1 - P_i)] \quad (5)$$

機械学習で避けられない過学習という問題に対して、提案手法では L2 正則化を採用した。最後に識別タスクの損失関数値 E_{det} と三つの補助タスクの損失関数値を次式で合計して、TCDCN 最後の損失関数値とする。

$$E = E_{\text{det}} + E_{\text{eye}} + E_{\text{mouth}} + E_{\text{face}} + \|W\|_2^2 \quad (6)$$

ここで、 E_{eye} , E_{mouth} , E_{face} はそれぞれ目、口、顔の損失関数であり、 $\|W\|_2^2$ は L2 正則化関数である。

次にニューラルネットワークの機械学習部分について本研究が採用する方法は勾配降下法である。原理は次式で示す。

$$W = W_0 - nW'_0 \quad (7)$$

ここで、 W_0 は元パラメータ行列、 n は学習率である。

本研究学習率を 0.001 に指定した。

機械学習の結果について、学習 3 万回を設定し、4.1 節で説明したように 330 枚の学習データの内七割のデータを学習集合として、残り三割のテストデータに対して実験を行った。一部のテスト結果を図 7 に示す。



図 7 機械学習の実験結果例

5. キャラクター陰影付け

4 章で説明した手法により獲得した特徴点座標情報を通じて必要な特徴線を計算し、線形計画法を使って、陰影を付けたい目標区域を囲む。本手法では、アニメキャラクターの顔部分に限って特徴点を学習されたため、陰影付けはアニメキャラクターの顔の部分に限定している。

5.1. 陰影付け目標区域の形状・位置分析

適切なアニメキャラクターの陰影を生成するために、事前に現存の陰影付け済アニメキャラクターの陰影区域の位置と形状を分析した。

本研究がネット上の約 400 枚のアニメキャラクター画像の顔陰影付け区域をまとめて分析した結果、現在のアニメキャラクターの陰影付け区域はほぼ光源に応じて、顔の側面、目、眉、髪この四つの区域につけられるという結論に達した。

この分析結果と、提案手法で設定している特徴点の位置とを照らし合わせた結果、陰影付け実験は顔の側面、目、眉三ヶ所で行うこととした。以下、陰影付け位置と形状について、図 8 の例を用いて説明する。



図 8 アニメキャラクター例 [5]

最初は顔側面の陰影について、アニメの処理は図 9 のように目の近くから、光源と顔向きに応じて目から顎方面に陰影境界線を描くというパターンが多い。

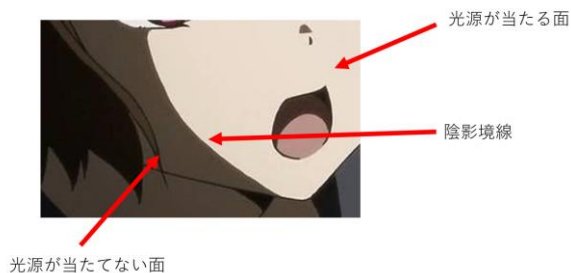


図 9 顔の陰影分析

目の陰影について、図 10 で示したように、アニメの中でキャラクターの目の陰影もしくは睫毛の陰影が眼球の上だけで付ける場合が多い。形状は眼球の最上部（上瞼）の所から少し下の所を上回る陰影境界線を描いて、眼球の最上部とこの陰影境界線を囲んだ区域に陰影を付ける。

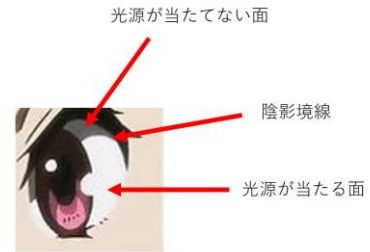


図 10 目の陰影分析

最後は眉の陰影について説明する。分析結果によれば、典型的な眉の陰影は図 11 で示した形である。他の顔上の陰影とは別に、眉の陰影は眉間という位置を表すために付ける場合が多い。そこで、眉の陰影は光源の変化に対して位置と形を変える程度が少ないという特徴がある。

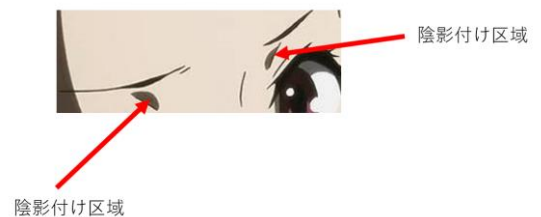


図 11 眉の陰影分析

5.2. キャラクターの陰影付け

前節で示した現存アニメキャラクター陰影の分析をベースにして、キャラクターの陰影付けを行う。

最も重要な特徴は、アニメキャラクターはほとんどの陰影輪郭が二次曲線として表すことができる、という点である。そこで提案手法では、すべての特徴線は、特徴点を通る二次曲線をあてはめて算出することとした。

4 章の特徴点検出実験の結果画像の中から図 12 の画像を選んで、陰影付けの実験を行った。理由として、この画像は検出結果が良い上、キャラクターの顔が他の物を隠れた部分がないということである。



図 12 陰影付け実験素材

5.2.1.顔の陰影付け

図 9 の分析をベースにして、キャラクターの目の近くから顎方面に一つ特徴線を計算し、それと顔輪郭線を囲む部分を顔の陰影とする。

図 5 の標記順を参照して、実験の第一歩はキャラクターの右顔の範囲を確定する。特徴点 1 番から 16 番を使って、顔輪郭線を計算する。そして、右顔と左顔を区別するため、顎の特徴点 7 番から 9 番、目の特徴点 18、19、20、23、27、28 番、鼻と口の特徴点 29 番から 35 番、眉の顔中心線に近い特徴点 40 番と 41 番全部 18 点を二次曲線あてはめて中心線を計算する。続いて、両目の下部分の特徴点 20、21、26、27 番を使って、下瞼線を計算する。最後は、キャラクター顔の範囲に限定するために、特徴点 1、17、22 でもう一つの二次曲線右顔線を計算する。この四つの特徴線によって、キャラクター右顔部分を確定できる。

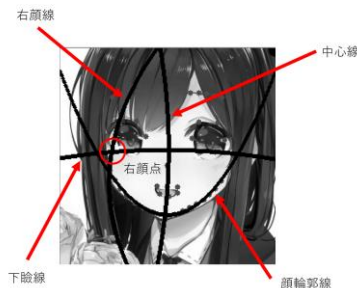


図 13 右顔特徴線

続いて、陰影の分析結果をもとにして、図 13 に示したように下瞼線と右顔線の交点（右顔点）近くから特徴点 7 番近くまでに三日月形の陰影を設計した。右顔点と特徴点 7 番の座標を通じて、この両点を繋がる直線の中心点を計算する。この中心点の xy 座標ともに 10 画素減少させ、中心点を元より顔の輪郭方面に移動し、右顔点と特徴点 7 とも同じ二次曲線あてはめを行って図 14 で示した右顔陰影線を計算する。10 画素という減少量は実験的に試行錯誤により設定した。



図 14 右顔陰影区域確定

さらに右顔陰影線、右顔線、顔輪郭線三つの特徴線を囲んだ区域を右顔の陰影区域とする。その陰影付け効果は図 15 で示す効果である。



図 15 右顔の陰影付け効果

5.2.2.目の陰影付け

二番目の陰影付けはキャラクターの目に対する陰影付けである。図 10 の分析結果で、現在のアニメキャラクターの目の陰影は上瞼の所から少し下の所を同じく三日月形に近い陰影を付ける。

図 5 の標記順を参照して、特徴点 23、24、25、28 番の座標情報を使った。そして、計算した特徴線の頂点をさらにキャラクターの目に適合させるために、目の最上端の特徴点 23 番と 24 番との中点を求め、微調整のために試行錯誤によりこの中点の y 座標を 10 画素増加させた。この中点と上記四つの特徴点から図 16 に示した左上目線と呼ぶ特徴線を求めた。

陰影の下部分について、特徴点 25 番と 28 番の中点を計算した。そして、計算する特徴線を上に凸の形状にするために、中点の y 座標を 10 画素減少させた。そこで、この中点、特徴点 25 番そして 28 番から左目陰影線を計算した。



図 16 左目特徴線

そして、右眼を同じパターンの処理を行って、特徴点 17、18、19、22 から右上目線と右目陰影線を計算する。最後で、上目線と目陰影線と囲んだ区域を目の陰影とする。その結果は図 17 の結果である。この図では、見やすくために、陰影の色を白に処理した。

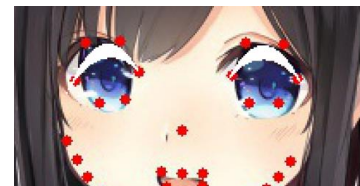


図 17 目の陰影付け効果

5.2.3.眉の陰影付け

最後は眉の陰影付けに関する部分である。同じく 5.1 章の分析で、図 11 を参照し、眉の下部分で一つの

区域に限定して、陰影を付ける。

左目の眉を例として、図 5 の標記順を参照して、本研究は特徴点 41 番から 45 番の座標情報を利用し図 18 のように左眉線を計算した。

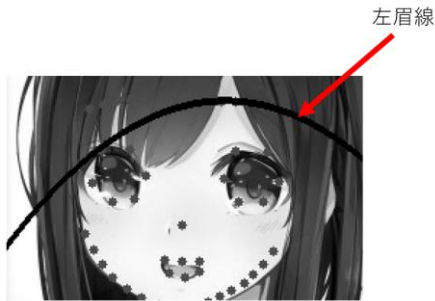


図 18 左眉特徴線

眉の陰影付け位置については、眉と目間の黄金分割点を利用することにした。同じく左目を例として、眉位置の特徴点 41 番から 45 番まで一つずつ左目上部の特徴点 23 番と 24 番の距離を計算した。全部の距離データの平均値を計算して、眉と目との間の距離とする。続いて、左眉線を y 軸の正方向に眉と目との間の距離に黄金分割比である 0.318 を乗じた数値分の距離に移動し、現在の左眉線の二次曲線方程式の係数を確定させる。続いて、もう一回左眉線を同じく y 軸の正方向を微小画素分移動し、二次曲線式を確定する。移動量は、試行錯誤の結果により図 19 の例では 2 画素とした。

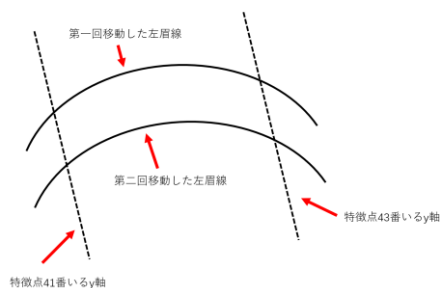


図 19 眉陰影範囲略図

最後、図 19 のように二回移動した左眉線、特徴 41 番の y 軸、そして特徴点 43 番の y 軸この四つの特徴線で囲んだ区域を左眉の陰影範囲とする。同様に右眉の陰影も付けた。その結果を図 20 に示す。



図 20 眉の陰影付け効果

6. おわりに

6.1. まとめ

本研究では、機械学習を導入し、ユーザーが 2D キャラクターを描く際の陰影付け支援手法を提案した。陰

影付けの第一歩として、本研究は機械学習の画像認識能力を通じて、アニメキャラクターの顔輪郭、口、鼻、目、眉の位置と形状を表す 45 点の特徴点をプログラムで自動的に求めた。

次に、獲得した位置情報を分析して、設定したキャラクターの陰影生成区域を確定し、陰影を付けた。

実験結果から分析すると、提案手法を用いるとイラストレーターは容易な操作により、自動的に一定品質のアニメキャラクターの顔陰影効果を生成できる。これによって手描きアニメ制作効率の向上が期待できる。

6.2. 今後の課題

陰影付けは特徴線と直接関わるため、機械学習でアニメキャラクターのフィッティング結果の精度は本研究の中で最も重要なポイントである。アニメキャラクターを対象とした理由の一つは、イラストのキャラクターに比べて量産されるアニメキャラクターの陰影付けはクオリティが低いという現状がある。今回の実験での抽出結果精度はアニメキャラクターに対してであれば許容範囲内と考えられる。今後、陰影付けのクオリティを向上し、イラストキャラクターに対しても適用させることは重要な課題である。

文 献

- [1] Sun, Y., Wang, X., Tang, X.: Deep convolutional network cascade for facial point detection, CVPR, pp. 3476–3483, 2013.
- [2] Zhanpeng Zhang, Ping Luo, Chen Change Loy, Xiaoou Tang. Facial Landmark Detection by Deep Multi-task Learning, Proceedings of European Conference on Computer Vision (ECCV), 2014.
- [3] Ultraist. Perl でアニメ顔を検出&解析する Imager::AnimeFace. Hatena Blog, 2009. <http://ultraist.hatenablog.com/entry/20090412/1239497216>
- [4] Nagadomi. lbpcascade_animeface. Github, 2009. https://github.com/nagadomi/lbpcascade_animeface
- [5] 京都アニメーション, 氷菓 (2012)