

2022年度 卒業論文

画像処理アルゴリズムを用いた
シェーダーパラメータ設定の自動化

指導教員：渡辺 大地 教授

メディア学部 ゲームサイエンスプロジェクト
学籍番号 M0119327
荒木 恒一

2023年2月

2022年度 卒業論文概要

論文題目

画像処理アルゴリズムを用いた
シェーダーパラメータ設定の自動化

メディア学部

学籍番号：M0119327

氏名

荒木 恒一

指導
教員

渡辺 大地 教授

キーワード

シェーダー、画像処理、3DCG、
自動化、特徴点マッチング、ヒストグラム比較

3DCG を描画する際に表面の質感や陰影の表現をはじめとする、3D モデルの「見え方」に大きく影響するものがシェーダーである。シェーダーは見え方や描画といったデザイン的な技術でありながら、プログラミング技術を必要とし、アルゴリズムで扱う難易度が高いという側面を持つ。そのため、グラフィックデザイナーやアーティスト自らが「見え方」を変更するために、描画のプログラムを改変することは、非常に困難である。そこで、シェーダーを含むプログラミングの予備知識のないデザイナーでも、描画の変更をできるようにするために、本研究では画像処理アルゴリズムを用いて自動化する手法を研究した。

本研究では数値調整に着目し、シェーダーその物を使いやすくするという方針ではなく、特徴点マッチングとヒストグラム比較の2つのアルゴリズムを用いて、再帰的手法を用いて4つの設定されたシェーダーの数値の自動で数値調整を行う。両アルゴリズムともシェーダーの影に着目して選んだが特徴点マッチングでは影の形状と段階的な影の濃さに、ヒストグラム比較では影の濃さと全体の影の面積に各々注力できるのではないかと考えた。この手法を用いて2つを比較運用することで処理速度や各々の特徴のデータを取り、そこから1つのアルゴリズムでは広いパターンに対応できないことと、ぼかし値が特徴点マッチングを運用するうえでは課題になることが分かった。

目次

第 1 章	はじめに	1
1.1	研究の目的	1
1.2	論文構成	4
第 2 章	研究手法	5
2.1	比較方法	5
2.1.1	特徴点マッチング	5
2.1.2	ヒストグラム比較	6
2.2	調整する数値とシェーダー	7
2.3	再帰的な数値調整	10
第 3 章	検証結果と考察	12
3.1	各手法の実行準備	12
3.2	実行時のデータ	15
3.2.1	特徴点マッチングを用いたパターン 1	15
3.2.2	ヒストグラム比較を用いたパターン 1	17
3.2.3	特徴点マッチングを用いたパターン 2	18
3.2.4	ヒストグラム比較を用いたパターン 2	20
3.2.5	特徴点マッチングを用いたパターン 3	21
3.2.6	ヒストグラム比較を用いたパターン 3	23
3.3	各パターンとアルゴリズムからの考察	24
3.3.1	パターン 1 からの考察	24
3.3.2	パターン 2 からの考察	26
3.3.3	パターン 3 からの考察	27
第 4 章	まとめ	29
	謝辞	30
	参考文献	31

目次

1.1	シェーダーパイプラインの図	2
1.2	複数のシェーダーの見え方の違い	2
2.1	算出された特徴点のサンプル	5
2.2	算出されたヒストグラム化の例	6
2.3	ヒストグラム比較の例	7
2.4	マスク値の最少と最大による変化	8
2.5	範囲値の最少と最大による変化	8
2.6	ぼかし値の最少と最大による変化	9
2.7	コントラスト値の最少と最大	10
3.1	手動設定した目標となるパターン 1	13
3.2	手動設定した目標となるパターン 2	14
3.3	手動設定した目標となるパターン 3	15
3.4	パターン 1 で特徴点マッチングを用いた誤差値が最少時の描画	16
3.5	パターン 1 で特徴点マッチングを用いた誤差値が最大時の描画	17
3.6	パターン 1 でヒストグラム比較を用いた誤差値が最少時の描画	18
3.7	パターン 1 でヒストグラム比較を用いた誤差値が最大時の描画	18
3.8	パターン 2 で特徴点マッチングを用いた誤差値が最少時の描画	19
3.9	パターン 2 で特徴点マッチングを用いた誤差値が最大時の描画	20
3.10	パターン 2 でヒストグラム比較を用いた誤差値が最少時の描画	21
3.11	パターン 2 でヒストグラム比較を用いた誤差値が最大時の描画	21
3.12	パターン 3 で特徴点マッチングを用いた誤差値が最少時の描画	22
3.13	パターン 3 で特徴点マッチングを用いた誤差値が最大時の描画	23
3.14	パターン 3 でヒストグラム比較を用いた誤差値が最少時の描画	24
3.15	パターン 3 でヒストグラム比較を用いた誤差値が最大時の描画	24
3.16	パターン 1 の実行結果比較	25
3.17	パターン 2 の実行結果比較	26
3.18	パターン 3 の実行結果比較	27

第 1 章

はじめに

1.1 研究の目的

コンピューター上での 3DCG の描画という技術は、ただの骨組みであるフレームを描画するところ [1] から歴史が始まった。映画やゲームといった鑑賞型の娯楽と、科学や医療といった検証的な二面 [2] から進化を重ね、今日まで様々な技術 [3] を吸収してきた。特にゲーム、医療分野 [4] での発展はすさまじく、様々な描画 [5] ができるようになった反面、今日では 3D モデルをディスプレイに描画するまでの一連の処理のことをレンダリングと呼ぶ [6]。

レンダリングとは、複数のシェーダー処理が行われるレンダリングパイプラインという処理を行うことになる。図 1.1 にレンダリングパイプラインの処理を簡潔にフローチャート化したものを示す。図 1.1 のレンダリングパイプラインは、大きく 4 つの処理に分けられる。第 1 のバークスシェーダーは、モデルの頂点ごとの処理のことである。描画するために構築された 3D 空間内での 3D モデルの頂点座標の入出力を行っている。第 2 のジオメトリシェーダーは、プリミティブごとの処理のことである。仮想的なカメラを元のディスプレイに描画するための 3D モデル頂点を増減し、頂点を線や面に加工することでカメラでディスプレイに描画された空間へと展開する。第 3 のラスタライザーは、処理したデータをディスプレイに表示できるようにする処理 [7] のことである。線形補間が行われた面や線をカメラ位置からディスプレイに表示する。第 4 のピクセルシェーダーは、ピクセル単位で行う特殊効果の処理のことである。画面上のピクセルの色を変更し、質感や影の表現を行う。この他にもテッセレーションやフラグメントシェーダー等が存在する。

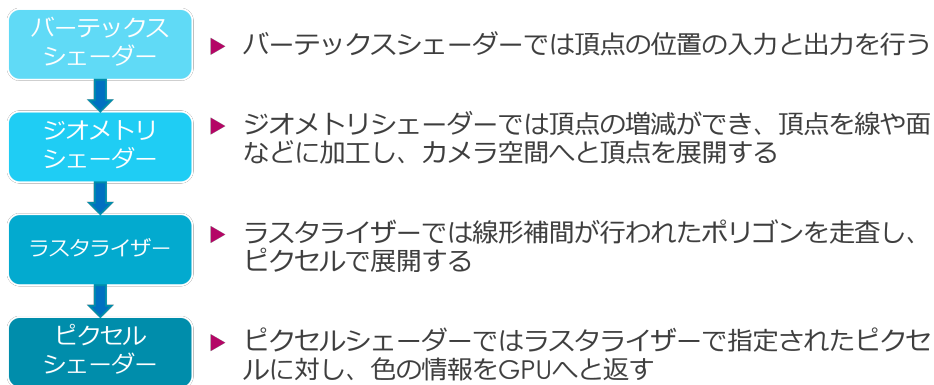


図 1.1 シェーダーパイプラインの図

シェーダーとは前述した通り、3D モデルに対して処理を行うアルゴリズムの事である。基本的に光源をモデルに対して照射し、照射された光によって生まれた影の描画と、マテリアルという質感にかかわる部分の計算を行う機能の総称となる。昨今のゲームではリアリティを重視したレンダリングとアニメーション調の描画を行うトゥーンシェーダーの2つが多く見られる。図 1.2 にシェーダーの違いによるモデルの見え方の違いを示す。シェーダーは図 1.2 のように対象となるモデルの質感によって計算式や処理の負荷が大きく変わるもの [8] であり、ゲームにとって重要な要素である「見た目」にかかわる。ゲーム開発の現場ではシェーダーの実装は終盤になりがちなもの、工数は多く設けられやすく、微調整などで時間がかかるものである。

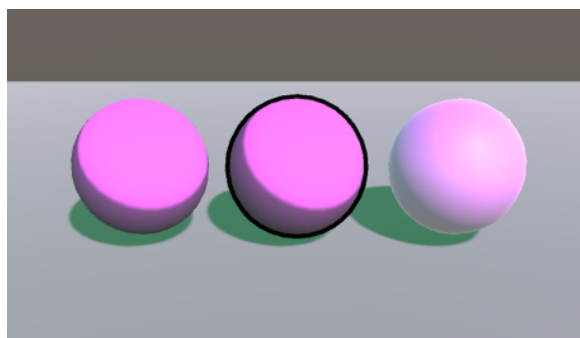


図 1.2 複数のシェーダーの見え方の違い

シェーダーはアルゴリズムとしてのコーディング難易度が高いという点に付随してゲーム開発では大きな問題点がある。それはシェーダーはグラフィックに大きくかかわる部分を扱う点にあ

る。シェーダーはアルゴリズムとしての実装難易度が高いだけでは問題点にはならないが、ゲームの開発においてグラフィックを扱うのがアーティストであるため問題となる。

モデルの描画は、ユーザーがゲームをプレイする上で視覚という大きなリソースが割かれるものであるため、実装する際により影や質感といった所の微細な調整や、影そのものでの演出などが多く用いられる。前述の通りシェーダーはその技術の進歩 [9] とハードウェア面 [10] での進歩が重なり、複雑化の一途を辿った。ゲーム開発の現場ではグラフィックアーティストとプログラマーの橋渡しの存在として、テクニカルアーティストという職種が存在するほどに、専門的な知識 [11] がなくては実装が困難になっていった。そのため、技術の進歩と共に美しいグラフィックを映し出せるようになったレンダリング技術は、グラフィックアーティストが簡単に操作できるものではなくなってしまった。この問題点を解決しなくては 3DCG の実装はハードウェアの機能とレンダリング技術の発展と共に専門的な知識がなければ実装が困難になっていってしまう。

そのため、本論ではレンダリングのコーティングそのものの自動化ではなく、調整する部分に着目して簡略化を目標とした。レンダリングの簡略化というのは多くの先行研究がなされている。Seitz ら [12] はメタプログラミングという手法を用いて簡略化を目指した。松尾ら [13] は、リアルタイムレンダリングでの輪郭線の描画を誇張化した。しかしながら、誰でも使いやすいシェーダーを作成する、実装の動作を軽くする、多機能な 1 つのシェーダー [14] を目標にするという 3 点が特に多く、シェーダに対して直接的なアプローチの研究が多い。

これらの研究ではシェーダーそのものに手を加え、簡略化する研究が多かったが、本研究ではシェーダーそのものではなく外部からの数値調整に着目して実装を行った。グラフィックアーティストがプログラミングを実装する際に行う数値の調整に着目した。シェーダーは基本的に描画したい質感や陰影によって異なる基盤となる計算式 [15] が存在し、それらを 3D モデルとマテリアルごとに調整を行う。開発現場でのシェーダーの実装は元となる画像やイメージをもとにそれを 3DCG へと落とし込んでいく際、ライトの反射の関係で、3DCG とイメージが乖離 [16] することがあるため、本研究では元画像に近似した描画を自動で行うというのを目指した。そのた

め、本研究ではグラフィックアーティストの望むアーティスト自身の考えるイメージを手描きの絵やサンプル画像といった形で出力したものを元画像とする。その元画像に近い描画を行う為に既存のシェーダーの数値調整を画像処理アルゴリズムによって自動で行い、簡略化するというのを目的に据えた。本研究では、シェーダー使用の難化を問題視し、その問題を解決するために画像処理アルゴリズムを用いて自動化の検討を行った。特徴点マッチングを用いて図形に注目した画像比較と、ヒストグラム比較で画素数に基づいた画像比較を行う。これら2つの画像処理アルゴリズムを用いて、再帰的手法でパラメータ設定を自動化して本研究の目的に沿うアルゴリズムを検証する。本研究では目標を設定し、再帰的手法を実行することでどれだけの誤差が生まれるかを測定した。アルゴリズムごとに大きく得手不得手の違いが表れ、複合させることでより良い手法になると仮説を立てた。しかし、精度はまだ低く、実行時間も長いという結果に終わった。

1.2 論文構成

本論分は全4章にて構成する。2章では研究手法について述べ、3章では実行結果から考察を行う。そして、4章にてまとめを述べる。

第 2 章

研究手法

本研究ではコーティングの知識がないアーティストがシェーダーを扱うために、シェーダーの調整を自動化する手段として画像処理アルゴリズムを 2 つを比較する。

2.1 比較方法

本研究では 2 つのアルゴリズムを用いて、画像処理の正確さや処理の速度、類似度の設定などから「使いやすい」アルゴリズム [17] を比較検討する。

2.1.1 特徴点マッチング

初めに特徴点マッチングに関して説明する。特徴点マッチング [18] の大まかな説明としては、2 つの画像で特徴点と呼ばれる画像の回転やスケールの変更に強い点を算出し比較するという画像処理アルゴリズムである。図 2.1 は実際に特徴点として算出された部分に印をつけた画像である。

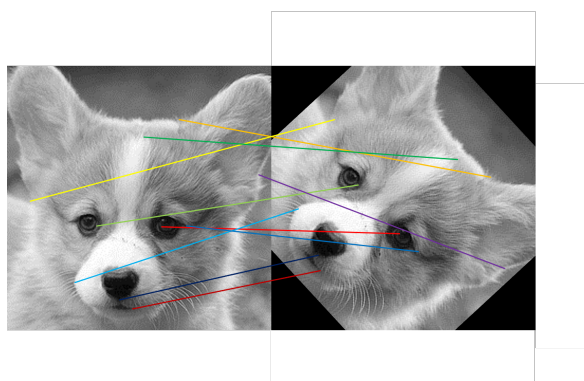


図 2.1 算出された特徴点のサンプル

特徴点マッチングでは特徴点と呼ばれる、画像が拡大縮小されても全体的な比率が変わらない

点同士の距離感などを抽出して画像の大枠をデータ化し、それらを比較計算することで類似度が算出される計算方式である。このアルゴリズムを用いるメリットとしては画像の変化に強い特徴点を算出するため、画像のサイズなどが変わっても比較に対応できる事である。デメリットとして特徴点の距離で類似度を計算するため、画像の色合いに対して対応することができない。加えて処理にかなり時間がかかり、2つの画像を1回比較するだけで、1秒前後かかってしまう。

2.1.2 ヒストグラム比較

次にヒストグラム比較 [19] と呼ばれる画像処理アルゴリズムに関して説明する。ヒストグラム比較というのは、2つの画像で画素値の個数をヒストグラム化し、それを比較する画像処理アルゴリズムである。図 2.2 は画像を画素数によって実際にヒストグラム化したものを示す。

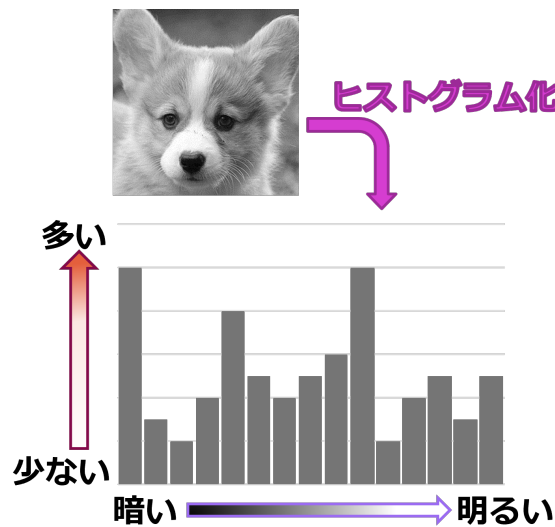


図 2.2 算出されたヒストグラム化の例

図 2.3 は画像をヒストグラム化して比較したものを示す。

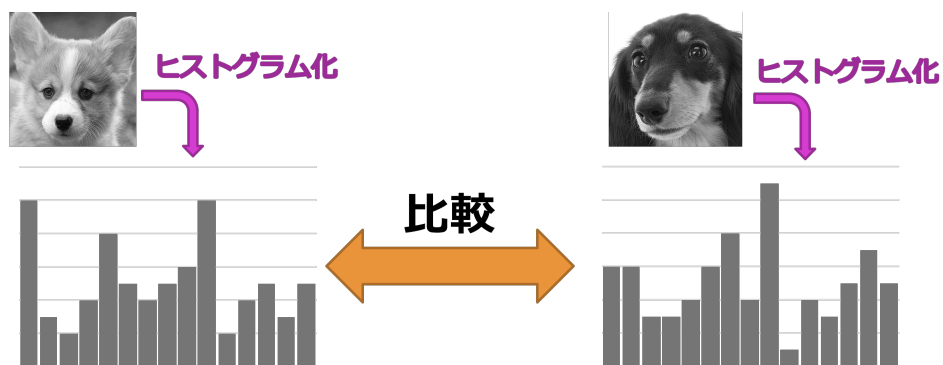


図 2.3 ヒストグラム比較の例

メリットとしてあげられるのは処理が速く、ヒストグラムの比率を比較するため画素が荒かったり大まかな画像でも色合いが一緒であれば類似画像として識別することができる。デメリットとして全体の画素数が同じ比率であれば類似度が高い数値になってしまう為、人間の認識よりも類似度が高く出てしまう。これら 2 つアルゴリズムで比較を行い、この研究の目的であるグラフィックデザイナーがシェーダーを扱うのを簡略化するという目的に最も適したアルゴリズムを考察する。

2.2 調整する数値とシェーダー

2 つのアルゴリズムそのものでは数値調整の自動化とともに画像を近似させることはできない。そのため、本研究では再帰的手法を用いる。まず、前提として今回の実験で使うシェーダーにはマスク値、範囲値、ぼかし値、コントラスト値の 4 つのパラメーターを変化できるように設定した [20]。マスク値は影の強度を変化させる値であり、主に影になっている部分の明暗に関わる。図 2.4 にはマスク値の最少と最大による変化を示す。

マスク値(影の強度)

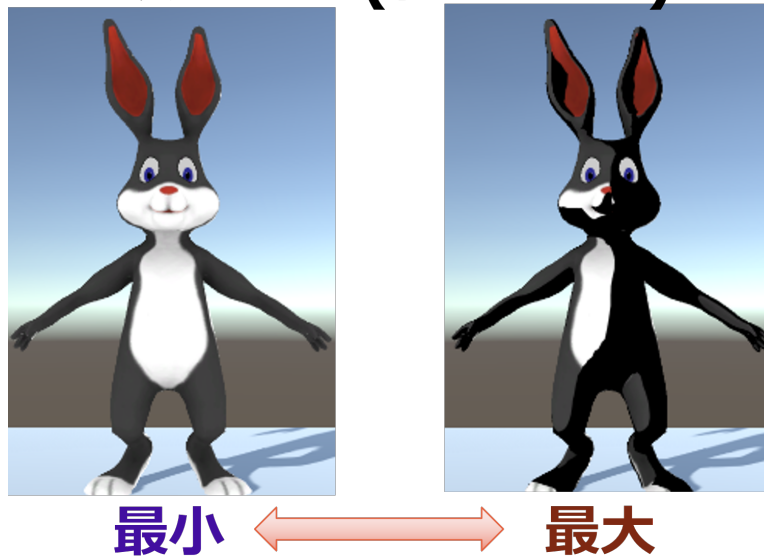


図 2.4 マスク値の最少と最大による変化

範囲値は影の範囲を増減させる値で、同じ光源でも影のかかる面積を変化させる。図 2.5 には範囲値の最少と最大による変化を示す。

範囲値(影の範囲)

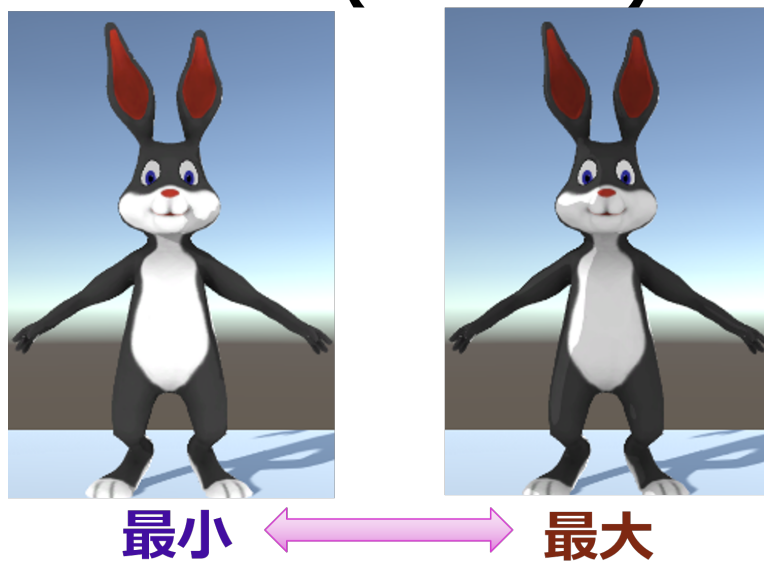


図 2.5 範囲値の最少と最大による変化

ぼかし値は影の見え方を変化させる値で、0の場合陰影に強いメリハリがつく。図 2.6 にはぼかし値の最少と最大による変化を示す。

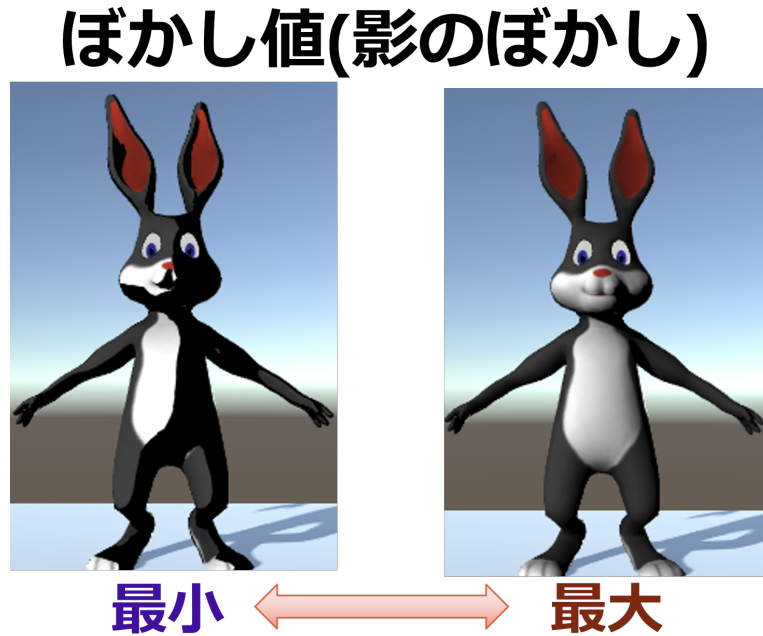


図 2.6 ぼかし値の最少と最大による変化

コントラスト値は影のつき方を変化させる値であり、光源からかかる影の強さにかかわる。図 2.7 にはコントラスト値の最少と最大による変化を示す。

コントラスト値(明暗の差)

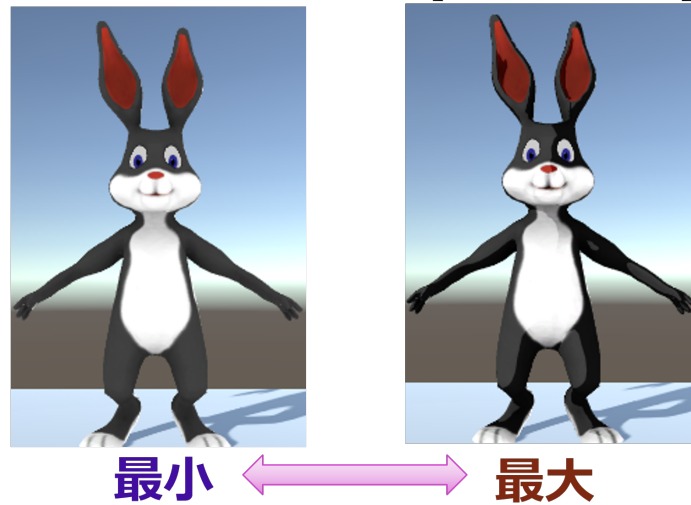


図 2.7 コントラスト値の最少と最大

各々 0~100 の間で数値変更を行えるようになっており、初期数値は 0 とした。

2.3 再帰的な数値調整

最初に描画したいイメージの元画像と元画像に似せたいシェーダーの描画を準備し、その元画像にどこまで似せるかを指定する目標類似度を設定する。次に元画像と現在のシェーダーパラメータを用いた描画状況をスクリーンショットした画像をアルゴリズムを用いて類似度を算出し、目標類似度に到達していれば終了する。目標類似度に到達しなかった場合はシェーダーの 4 つのパラメーターから 1 つをランダムに指定し、数値を 5 増加させるという処理を行う。パラメーター増加後、もう一度スクリーンショットを撮り、元画像と類似度を算出し類似度が上がっていれば増加した数値を保持したまま続行し、類似度が下がっていれば増加させた数値を戻す。目標類似度に到達すれば終了、到達しなければもう一度パラメーターの増加を行い、元画像とパラメーター増加後のスクリーンショットの比較を行うというループを行う。

数値の設定はランダムで増加することになるが、類似度という指標に従って必要な数値を増加

させるため、最終的に目的の数値に収束していく。しかしこの手法では、類似度が頭打ちになってしまい上がらなくなる場合や設定された目標類似度が高すぎる場合、半永久的にループを抜けられなくなってしまう。そのため例外的な処理として、20回類似度が増加しないループをした場合強制的に終了させる。

第 3 章

検証結果と考察

3.1 各手法の実行準備

本実験では初めに元画像を見ながら手動でパラメータを調整した理想値を設定する。この理想値に対してどれだけ本手法で近づけるかで、本手法の精度判断を行う。4つのパラメータそれぞれの理想値と、本手法適用後の4つのパラメータをそれぞれで差分を取り、差分を合算した数値を以下「誤差値」と呼ぶ。誤差値は高いほど目標の基準から離れていることを表す。大きく描画の違う3パターンを準備し、それぞれのパターンにおいて両方のアルゴリズムで試行を行い、データを取る。今回の実験では試行回数、実行時間、誤差値を比較対象としてデータを取った。

表 1.1 に今回実験で使用する3パターンの各数値を示す。

表 3.1 実験で使う3パターンの各パラメータ値

	マスク値	範囲値	ぼかし値	コントラスト値
パターン 1	60	70	0	20
パターン 2	80	70	10	100
パターン 3	100	80	0	50

本実験において手動パラメータ設定をした精度測定対象の3パターンには、各々狙いが存在する。パターン1ではトゥーンシェーダーによる陰影のメリハリの強い描画を生成するために、ぼかし値とコントラスト値を低く設定した。これにより影の面積と陰影の差という部分の精度を計る設定を行った。図 3.1 にはパターン1として設定した描画を示す。



図 3.1 手動設定した目標となるパターン 1

パターン 2 では淡い目の影とコントラストの強い描画を設定し、薄い影がかかる場合少し暗くなっていることを測定する。図 3.2 にはパターン 2 として設定した描画を示す。



図 3.2 手動設定した目標となるパターン 2

パターン 3 では今回設定したパターンで最も強く影を描画し、影の暗さと面積、陰影のメリハリをかなり強めに設定した。これによって強い影がある場合影の当たる面積を正確に測れるか、影が完全な黒の場合の動作のデータを取る設定にした。図 3.3 にはパターン 3 として設定した描画を示す。



図 3.3 手動設定した目標となるパターン 3

これら 3 パターンを 2 つのアルゴリズムで各々 10 回ずつのデータを取った。

3.2 実行時のデータ

データを取った回数と、ループ回数、実行時間、誤差値を記載する。また、誤差値の最少と最大の時の描画をデータの下に乘せる。実行時間は小数点第 1 位まで記載する。

3.2.1 特徴点マッチングを用いたパターン 1

マスク値 60、範囲値 70、ぼかし値 0、コントラスト値 20 を特徴点マッチングで実行した全 10 回のデータである。

表 3.2 特徴点マッチングを用いたパターン 1

	ループ回数	実行時間 (秒)	誤差値
1 回目	63	92.2	30
2 回目	84	140.0	10
3 回目	58	85.5	20
4 回目	102	184.4	15
5 回目	55	78.4	10
6 回目	47	69.4	25
7 回目	84	112.9	0
8 回目	67	95.6	15
9 回目	113	183.3	55
10 回目	62	81.5	5

図 3.4 にはパターン 1 で特徴点マッチングを用いた誤差値が最少時の描画を示す。

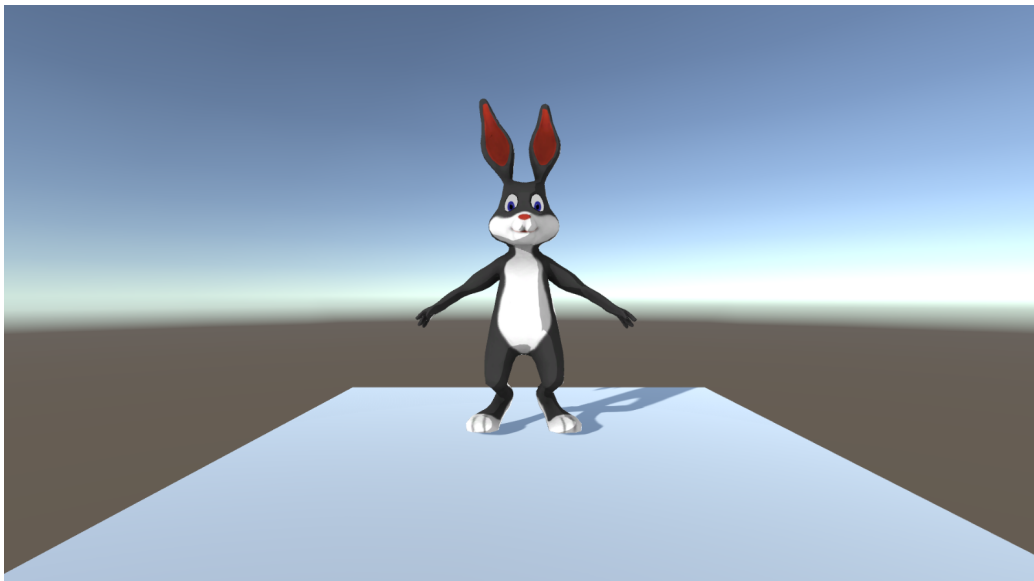


図 3.4 パターン 1 で特徴点マッチングを用いた誤差値が最少時の描画

図 3.5 にはパターン 1 で特徴点マッチングを用いた誤差値が最大時の描画を示す。

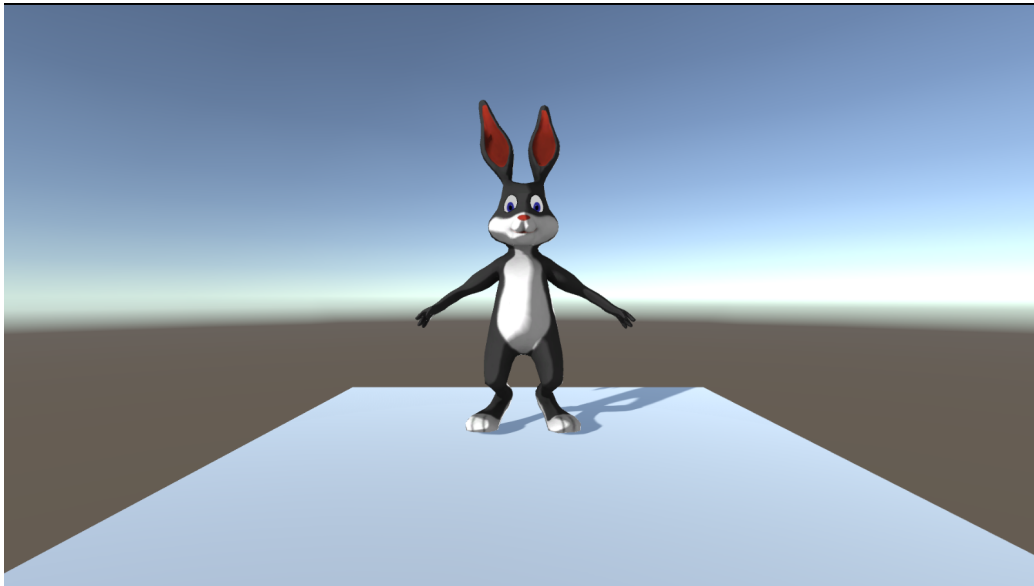


図 3.5 パターン 1 で特徴点マッチングを用いた誤差値が最大時の描画

3.2.2 ヒストグラム比較を用いたパターン 1

マスク値 60、範囲値 70、ぼかし値 0、コントラスト値 20 をヒストグラム比較で実行した全 10 回のデータである。

表 3.3 ヒストグラム比較を用いたパターン 1

	ループ回数	実行時間 (秒)	誤差値
1 回目	35	19.1	25
2 回目	46	22.5	40
3 回目	76	34.3	25
4 回目	62	36.7	15
5 回目	59	24.9	10
6 回目	104	41.1	5
7 回目	78	25.0	65
8 回目	66	27.4	45
9 回目	55	25.8	0
10 回目	83	30.4	5

図 3.6 にはパターン 1 でヒストグラム比較を用いた誤差値が最少時の描画を示す。

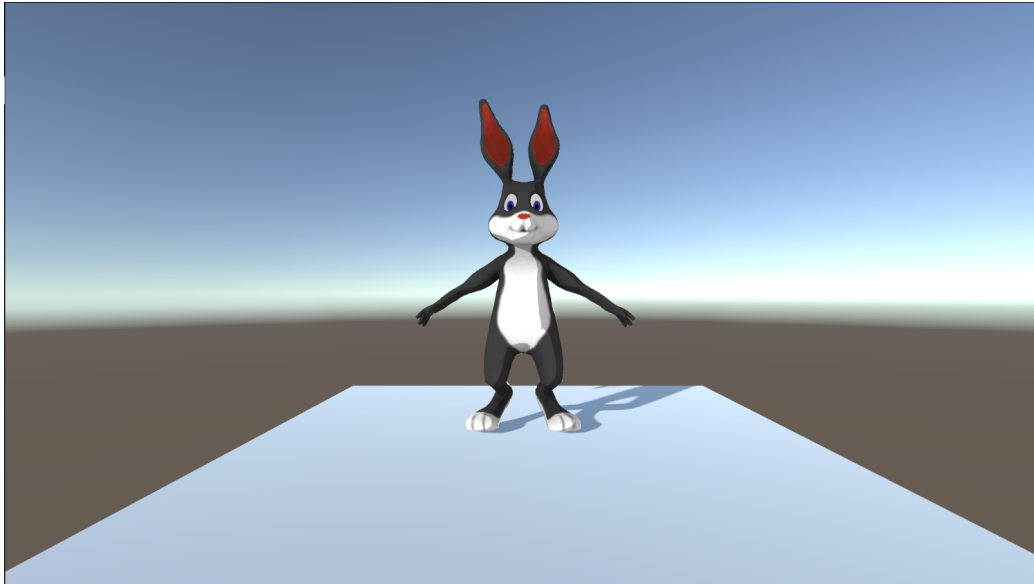


図 3.6 パターン 1 でヒストグラム比較を用いた誤差値が最少時の描画

図 3.7 にはパターン 1 でヒストグラム比較を用いた誤差値が最大時の描画を示す。

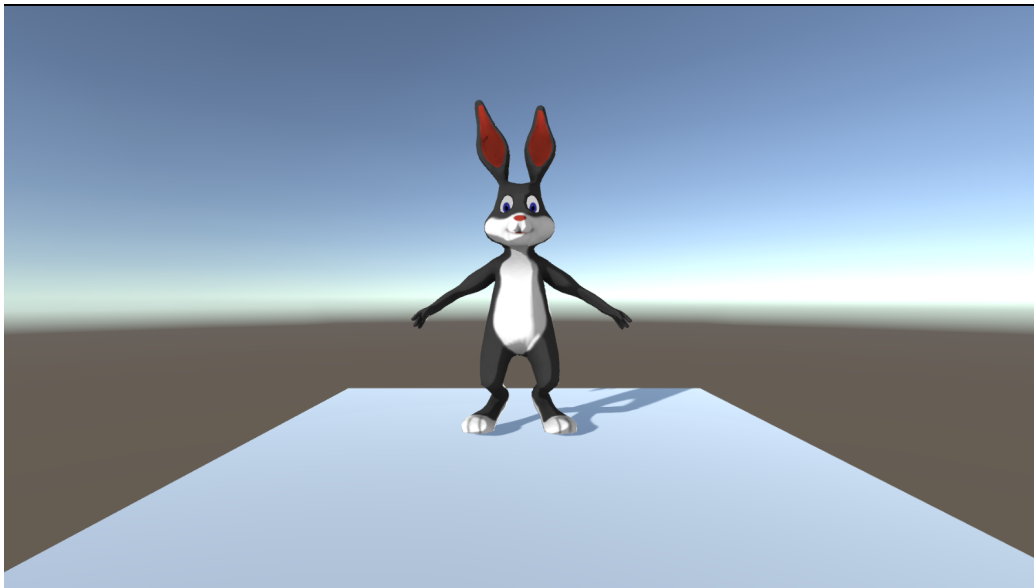


図 3.7 パターン 1 でヒストグラム比較を用いた誤差値が最大時の描画

3.2.3 特徴点マッチングを用いたパターン 2

マスク値 80、範囲値 70、ぼかし値 10、コントラスト値 100 を特徴点マッチングで実行した全 10 回のデータである。

表 3.4 特徴点マッチングを用いたパターン 2

	ループ回数	実行時間 (秒)	誤差値
1 回目	67	89.7	25
2 回目	78	114.6	20
3 回目	55	82.4	35
4 回目	105	167.3	15
5 回目	78	104.4	20
6 回目	60	89.5	40
7 回目	44	68.1	65
8 回目	122	192.2	55
9 回目	86	89.6	40
10 回目	90	132.8	20

図 3.8 にはパターン 2 で特徴点マッチングを用いた誤差値が最少時の描画を示す。

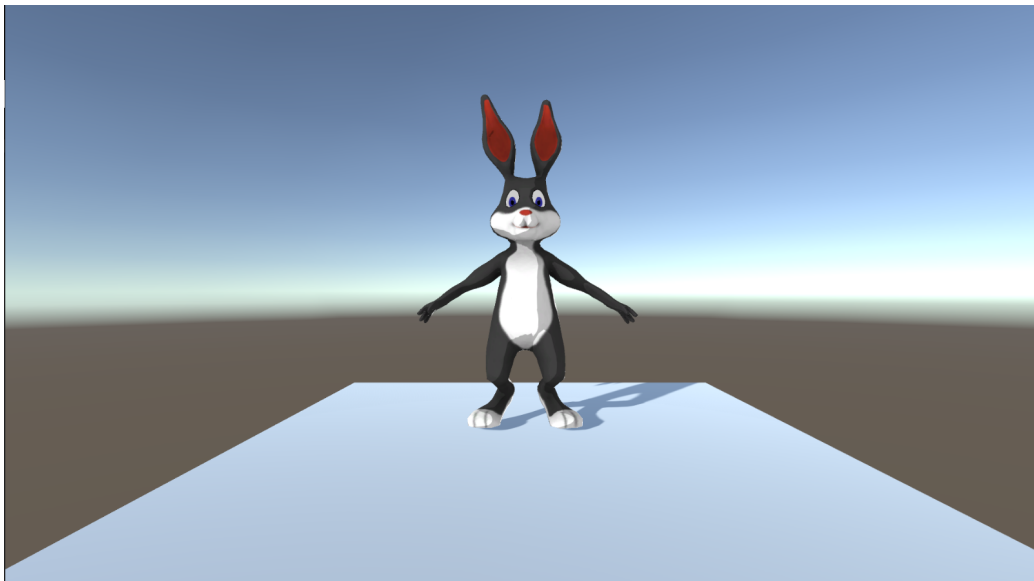


図 3.8 パターン 2 で特徴点マッチングを用いた誤差値が最少時の描画

図 3.9 にはパターン 2 で特徴点マッチングを用いた誤差値が最大時の描画を示す。

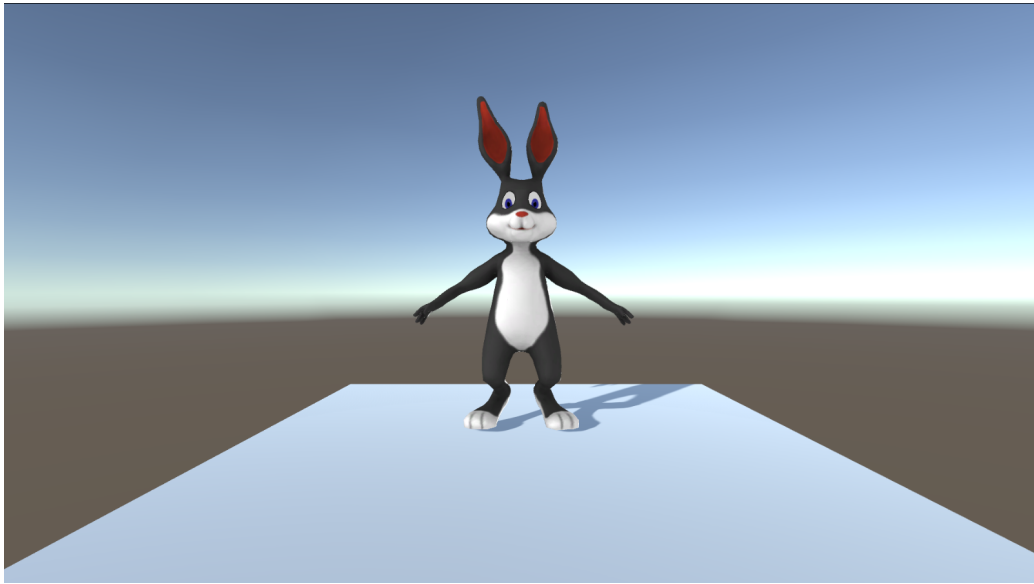


図 3.9 パターン 2 で特徴点マッチングを用いた誤差値が最大時の描画

3.2.4 ヒストグラム比較を用いたパターン 2

マスク値 80、範囲値 70、ぼかし値 10、コントラスト値 100 をヒストグラム比較で実行した全 10 回のデータである。

表 3.5 ヒストグラム比較を用いたパターン 2

	ループ回数	実行時間 (秒)	誤差値
1 回目	80	34.8	45
2 回目	44	15.6	10
3 回目	79	35.1	30
4 回目	64	27.7	5
5 回目	82	28.0	0
6 回目	83	30.7	15
7 回目	79	27.2	40
8 回目	61	23.8	15
9 回目	89	31.5	30
10 回目	65	27.9	25

図 3.10 にはパターン 2 でヒストグラム比較を用いた誤差値が最少時の描画を示す。

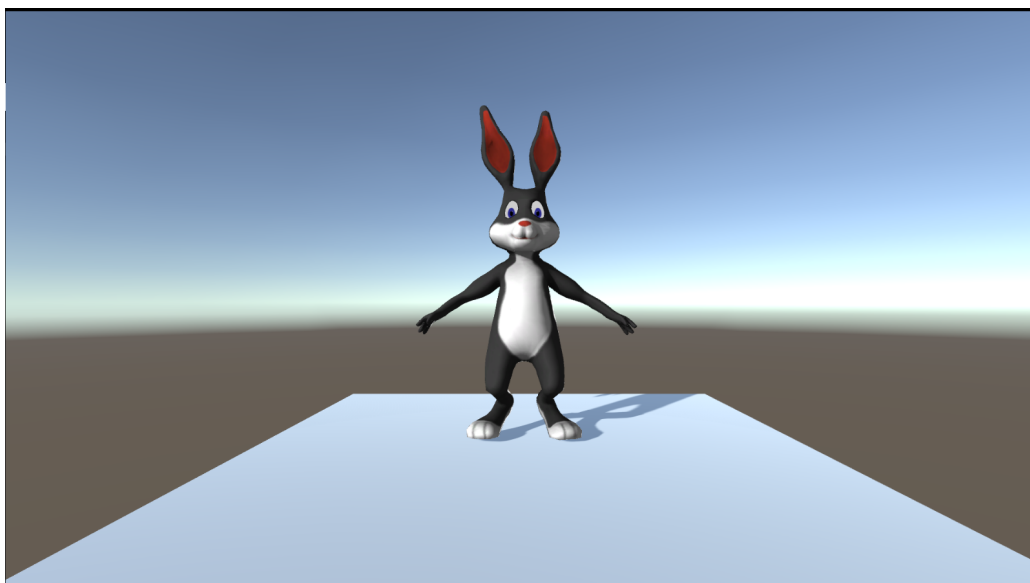


図 3.10 パターン 2 でヒストグラム比較を用いた誤差値が最少時の描画

図 3.11 にはパターン 2 でヒストグラム比較を用いた誤差値が最大時の描画を示す。

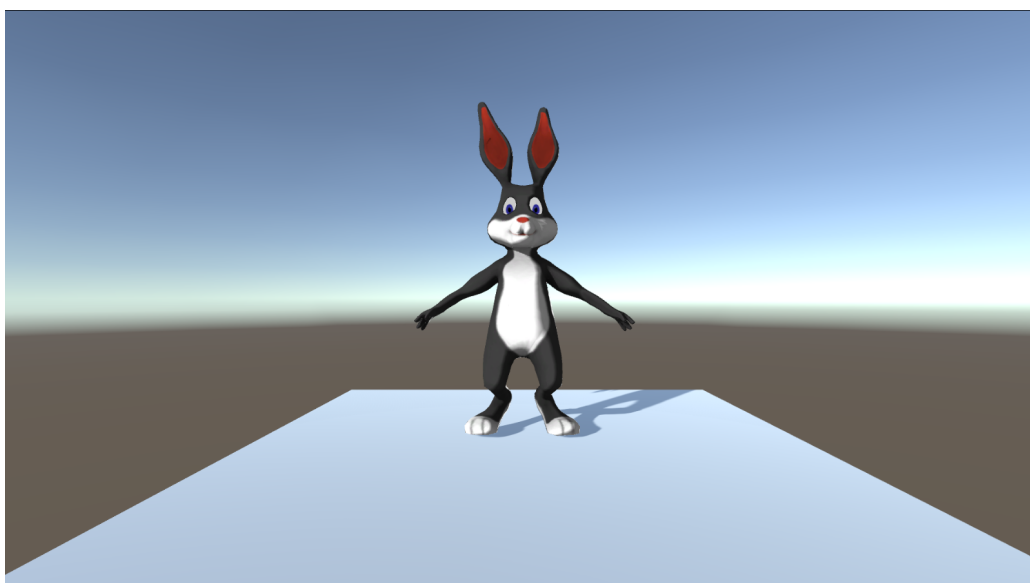


図 3.11 パターン 2 でヒストグラム比較を用いた誤差値が最大時の描画

3.2.5 特徴点マッチングを用いたパターン 3

マスク値 100、範囲値 80、ぼかし値 0、コントラスト値 50 を特徴点マッチングで実行した全 10 回のデータである。

表 3.6 特徴点マッチングを用いたパターン 3

	ループ回数	実行時間 (秒)	誤差値
1 回目	76	102.4	15
2 回目	69	88.6	30
3 回目	101	132.8	10
4 回目	78	106.1	45
5 回目	89	112.9	55
6 回目	108	164.4	20
7 回目	115	140.0	35
8 回目	96	111.3	15
9 回目	127	189.4	30
10 回目	72	91.0	10

図 3.12 にはパターン 3 で特徴点マッチングを用いた誤差値が最少時の描画を示す。

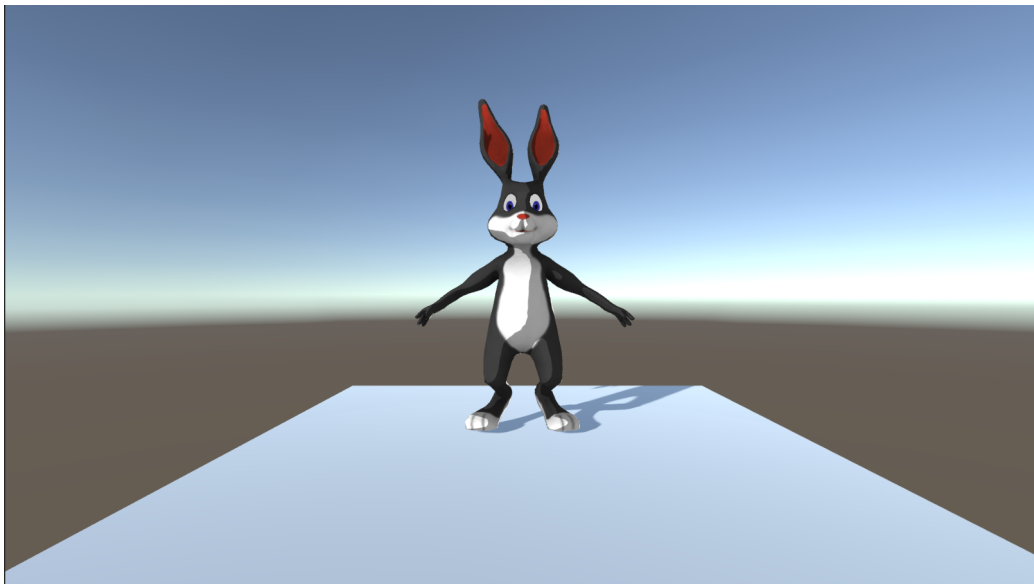


図 3.12 パターン 3 で特徴点マッチングを用いた誤差値が最少時の描画

図 3.13 にはパターン 3 で特徴点マッチングを用いた誤差値が最大時の描画を示す。

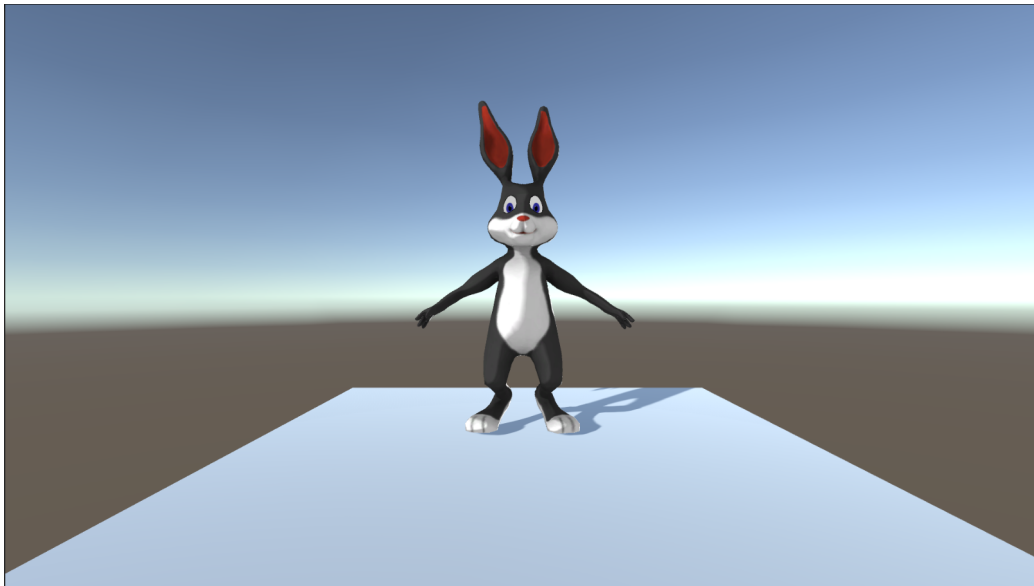


図 3.13 パターン 3 で特徴点マッチングを用いた誤差値が最大時の描画

3.2.6 ヒストグラム比較を用いたパターン 3

マスク値 100、範囲値 80、ぼかし値 0、コントラスト値 50 をヒストグラム比較で実行した全 10 回のデータである。

表 3.7 特徴点マッチングを用いたパターン 3

	ループ回数	実行時間 (秒)	誤差値
1 回目	114	29.3	10
2 回目	64	20.1	20
3 回目	89	31.9	25
4 回目	65	28.3	35
5 回目	78	19.7	20
6 回目	54	26.7	15
7 回目	75	27.4	30
8 回目	82	31.0	20
9 回目	103	28.6	45
10 回目	121	29.4	5

図 3.14 にはパターン 3 でヒストグラム比較を用いた誤差値が最少時の描画を示す。

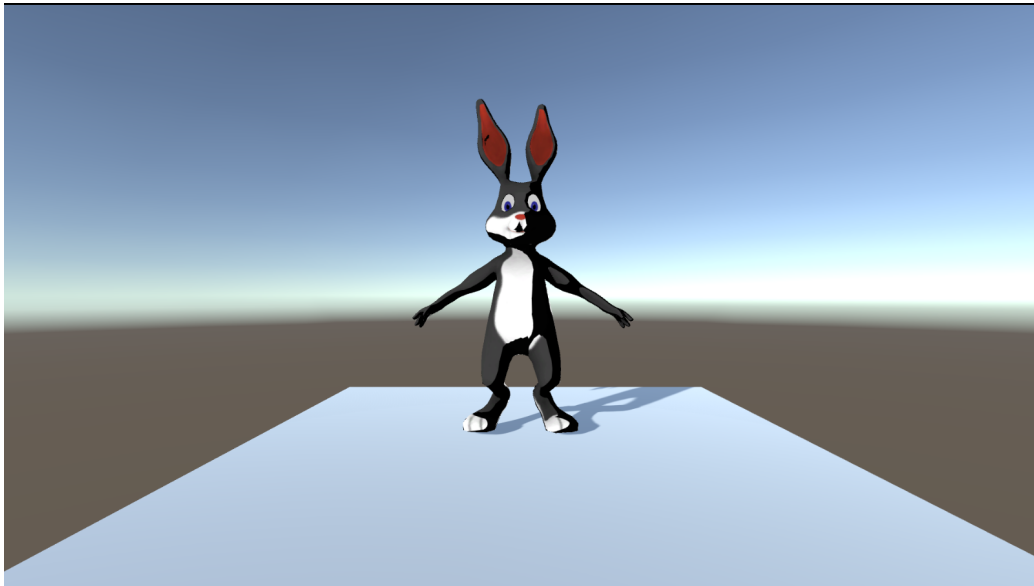


図 3.14 パターン 3 でヒストグラム比較を用いた誤差値が最少時の描画

図 3.15 にはパターン 3 でヒストグラム比較を用いた誤差値が最大時の描画を示す。

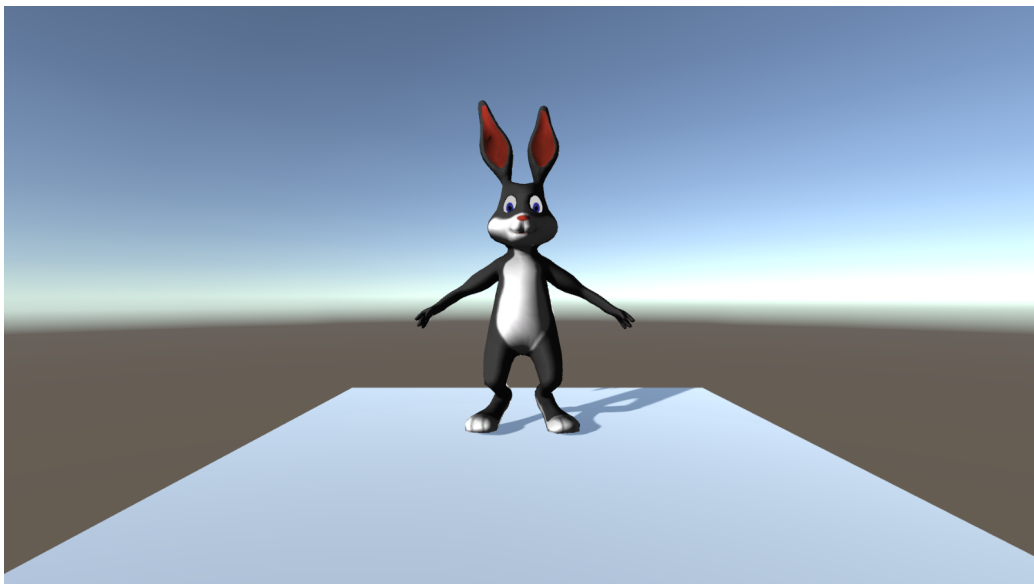


図 3.15 パターン 3 でヒストグラム比較を用いた誤差値が最大時の描画

3.3 各パターンとアルゴリズムからの考察

3.3.1 パターン 1 からの考察

図 3.16 にはパターン 1 で実行した二つのアルゴリズムでの結果を記す。

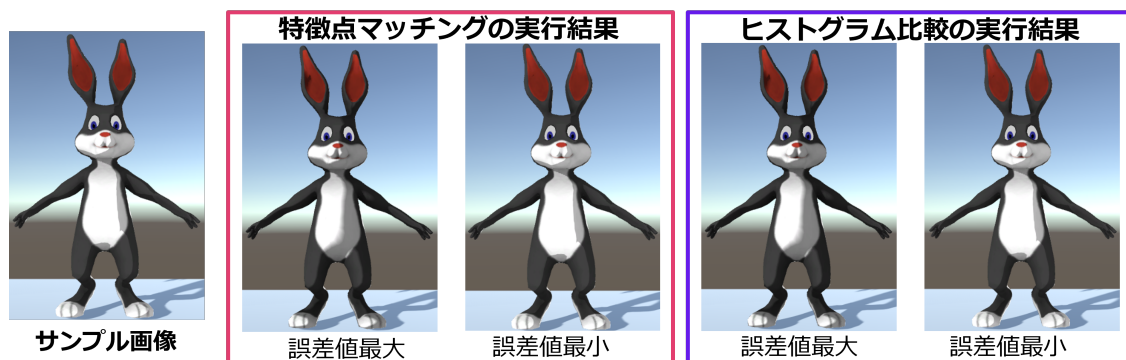


図 3.16 パターン 1 の実行結果比較

パターン 1 の実行結果からパターン 1 のぼかしがないトゥーンシェーダー特有の描画では影が強く特徴点を検出できるように、ヒストグラム比較を用いた場合よりも特徴点マッチングを用いた場合のほうが精度が高く出る場合が多かった。パターン 1 の特徴点マッチングのデータに着目した時、誤差値 20 以下の場合が 10 回中 7 回であり、このようなはっきりとした強い影を描画する場合、特徴点マッチングは適していると考えられる。しかし、実行時間に関しては 10 回中 1 分以内に収まる事が 1 度もないほど長く、想定していたよりも多くのループ回数がかかったという経緯もあるが、長い時では 1 ループ 1 秒以上もかかる結果になった。

対してパターン 1 のヒストグラム比較を用いたデータに着目した場合、誤差値はふり幅が広く、5 以下と 40 以上が各々 3 回以上も記録された。また振れ幅の大きかったデータを見るに、影の暗さという部分が影の濃さをマスク値を増加させるか影の強さであるコントラスト値を増加させるかで割れていた。この 2 つの値は光源から落ちる陰の計算に用いるコントラスト値と、落とされた影そのものの濃さを計算に使うマスク値で大きく違うものなのだが、ヒストグラム比較で算出される類似度では違いをあまり検出できなかった。その為、大きく誤差値が割れたのではないかという考察をした。また実行時間では特徴点マッチングと同じほどのループ回数でも 3~4 倍近くも実行時間に差ができ、画素数を単純に数えるヒストグラム比較である方が自動化における工数の削減という目的には適しているように考えられる。

3.3.2 パターン 2 からの考察

図 3.17 にはパターン 2 で実行した二つのアルゴリズムでの結果を記す。

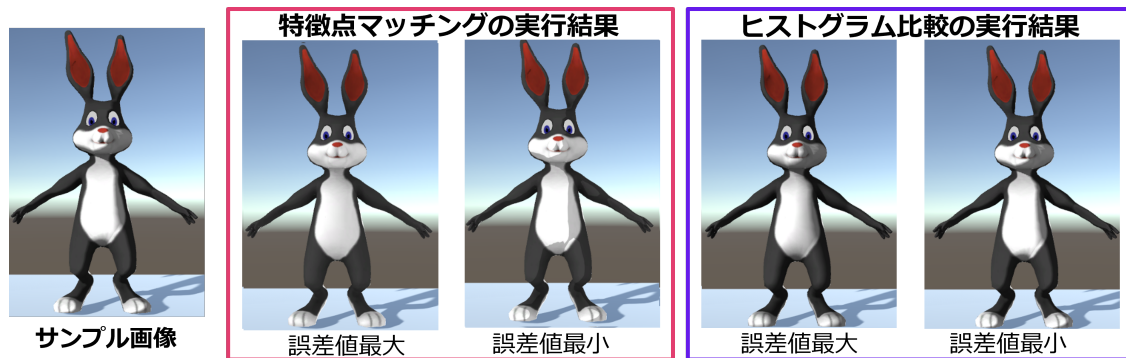


図 3.17 パターン 2 の実行結果比較

パターン 2 では最近のゲームで多く用いられるアニメ調のトゥーンシェーダーの使い方ではなく、ソフトな影を落とすタイプの描画を設定した。このパターン 2 に関しては顕著にシェーダーごとの特徴が出た結果になった。パターン 2 に特徴点マッチングを用いたデータではパターン 1 の場合よりも誤差値が全体的に悪い結果になった。パターン 1 とパターン 2 から考えるに、ソフトな影では影そのものに特徴点をとることが難しくなるのではないかと考えられた。特徴点では画像を図形としてみて画像の変化に強い特徴的な点を複数検出するため、ソフトな影では図形として認識できず、影の再現が難しくなるためこのような結果になったと考えられる。また、実行時間に関してもパターン 1 と変わらず大幅にかかってしまった。

パターン 2 のヒストグラム比較を用いたデータでは誤差値、実行時間ともに高い結果が出た。ソフトな影によって描画が行われる場合、影のかかっている部分は暗さがグラデーションのように描画される。このソフトな影がヒストグラム比較では画素ずつの判断になることで正確に図れていたため誤差値が良い結果になったと考えられる。ここから考えるに影に関連する色合いという要素ではヒストグラム比較が適していると分かる。また、ここまで高く出たのにはマスク値とコントラスト値が両方とも高い設定だったため、パターン 1 の時のようなふり幅が出なかったと

考えられる。

3.3.3 パターン 3 からの考察

図 3.18 にはパターン 3 で実行した二つのアルゴリズムでの結果を記す。

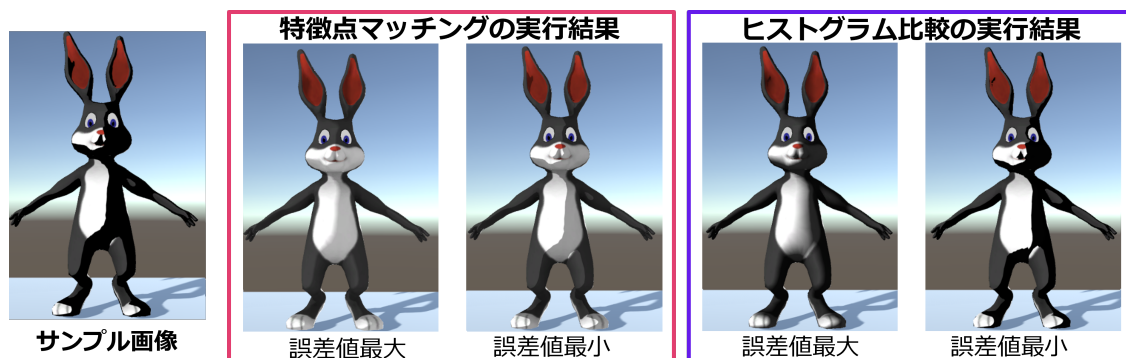


図 3.18 パターン 3 の実行結果比較

パターン 3 ではパターン 1 でトゥーンシェーダー特有のぼかしなしの描写に、強い影を落とす描写に設定した。パターン 1 では、特徴点マッチングは影のメリハリが強い描画への精度が高く、同じようにパターン 2 では色合いの変化に対してヒストグラム比較は適していた。そのため、明暗のメリハリが強い、色の濃い描画を設定し、比較を行った。

特徴点マッチングでパターン 3 を実行した場合、パターン 1 の時よりも誤差値の幅が広く、結果がかなりバラバラであった。特徴点マッチングはパターン 1、パターン 2 と同じく誤差値のぶれが大きい結果になった。これは陰影のメリハリが強いことで全体的に誤差値が高いが、パターン 3 では影の暗さがかなり強く、そこの差で誤差値の幅が広がったため、色合いの部分で大きく差がでたと考えられる。

またパターン 3 のヒストグラム比較を用いたデータでは安定して 20 前後の数値が多い結果となった。パターン 1 とパターン 2 から考えるに、パターン 1 のようにマスク値とコントラスト値の差が大きいことで数値が広く出たが、影の暗さが色合いに大きくかかわるためヒストグラム比較に適していると考えた。特徴点アルゴリズムとヒストグラム比較両方で各々の特徴が出たため

パターン 3 では最終的に全体の誤差値では一長一短の結果となった。

第 4 章

まとめ

本研究では 2 つのアルゴリズムを用いて比較実験を行った結果、それぞれのアルゴリズムにおける特徴をとらえることができた。特徴点マッチングでは陰影のメリハリが強いトゥーンシェーダー特有の描画での精度が高く、対して影の色合いといった部分に弱かった。ヒストグラム比較ではソフトな影に対して強く、トゥーンシェーダー的なメリハリのある描画には特徴点マッチングに一步劣るという結果になった。また、実行時間に関しては 2 つのアルゴリズムの違いが大きく出た。同じループ回数を行った際に特徴点マッチングでは 3~4 倍近くもの時間が発生した。実験前に考えていたよりも多くの時間がかかってしまったためこれ単体で実際に運用するのは難しいという結論に至った。また、今回はランダムで数値調整を繰り返したが、数値調整をより最適化することができればコーディングができない人でもシェーダーの数値調整ができるようになるのではないだろうか。

謝辞

東京工科大学メディア学部ではコロナ禍でなかなかキャンパスに通うこともできない中ではありましたが多くの学びを得ました。指導していただいた先生方に感謝を申し上げます。ありがとうございました。

また、研究室のメンバーには研究を進める中で様々な刺激や新しい視点を頂き、研究を進める過程でとても参考になりました。ありがとうございます。

また、研究の指導をしてくださった渡辺教授と阿部実験助手、論文の添削を手伝ってくださった吉田先輩の御三方には大変お世話になりました。ここで重ねてお礼申し上げたいと思います。何度も添削と査読をしていただき、研究、論文ともに客観的な視点と意見で研究を進めていただきました。御三方には研究の初期段階から何度も助けていただき、拙いながらもこの論文と研究を完成させることができたのはご指導くださる教授と先輩のおかげだったと考えております。ありがとうございました。

最後に、この研究にかかわった全ての方々に感謝を申し上げます。本当にありがとうございました。

参考文献

- [1] 小高文博, 佐藤建吉. Gpu コンピューティングの特徴と歴史. 年次大会, Vol. 9, pp. 14–16, 2014.
- [2] 鈴木雅隆, 周藤安造. ボリュームレンダリング. 可視化情報学会誌, Vol. 13, No. 49, pp. 102–106, 1993.
- [3] 周藤安造. レンダリングの手法 (iii) -医療応用を中心としたボリュームレンダリングの基礎から応用まで-. テレビジョン学会誌, Vol. 46, No. 4, pp. 490–496, 1992.
- [4] 周藤安造. 医用画像の三次元可視化. 日本機械学会誌, Vol. 96, No. 895, pp. 488–492, 1993.
- [5] 鈴木雅隆, 周藤安造. 複数 vm による並列分散システムのための動的な性能推定法の提案. マルチメディア通信と分散処理ワークショップ論文, Vol. 26, pp. 80–89, 2018.
- [6] 佐藤英司, 坂本雄児. ボリュームレンダリングの並列処理アルゴリズム. 映像情報メディア学会技術報告, Vol. 22, No. 9, pp. 61–66, 1998.
- [7] 小澤禎裕, 谷田川達也, 久保尋之, 森島繁生. 光学的最短経路長を用いた表面下散乱の高速計算による半透明物体のリアルタイム・レンダリング. 画像電子学会誌, Vol. 46, No. 4, pp. 533–546, 2017.
- [8] 河村容治. インテリア空間に適したラジオシティ法によるレンダリングの設定. 日本インテリア学会論文報告書, Vol. 17, pp. 9–20, 2007.
- [9] 周藤安造, 上野滋, 鈴木雅隆. 医学分野におけるボリュームレンダリングの原理と臨床応用. 可視化情報学会誌, Vol. 20, No. 1, pp. 175–178, 2000.
- [10] 山田泰永. ディープラーニング向け gpu 環境選択の実務. *Medical Imaging Technology*, Vol. 36, No. 2, pp. 76–80, 2018.
- [11] ソソラバラムバトタッチメグ, 藤本忠博, 千葉則茂. バイラテラルフィルタと多重サイズスプ

- ラットによるポイント表現された曲面の簡易化手法. 芸術科学会論文誌, Vol. 9, No. 3, pp. 140–153, 2010.
- [12] Kerry A. Seitz, Tim Foley, Serban D. Porumbescu, and John D. Owens. Staged metaprogramming for shader system development. *ACM Transactions on Graphics*, Vol. 38, No. 202, pp. 1–15, 2019.
- [13] 松尾隆志, 三上浩司, 渡辺大地, 近藤邦雄. リアルタイム 3dcg における物体の形状を考慮した輪郭線の誇張表現手法の提案. 芸術科学会論文誌, Vol. 10, No. 4, pp. 251–262, 2011.
- [14] 芝池祐星, 岩崎慶. 微分可能レンダリングを用いた髪の毛・毛皮の散乱パラメータ推定. 全国大会講演論文集, Vol. 84, pp. 255–256, 2022.
- [15] 笠原信一, 大久保寛. Gpgpu を活用した超高速 cg レンダリングのための効率的並列化アルゴリズム. 日本建築学会環境系論文集, Vol. 84, No. 759, pp. 543–552, 2019.
- [16] 岩崎慶, 中谷聡志, 吉本富士市. 自己反射を考慮した鏡面反射物体のリアルタイムレンダリング. 画像電子学会誌, Vol. 37, No. 4, pp. 405–411, 2008.
- [17] @ Best not best. Python + opencv で画像の類似度を求める. https://qiita.com/best_not_best/items/c9497ffb5240622ede0. 参照: 2022.9.1.
- [18] 富安史陽, 平山高嗣, 間瀬健二. Kalman-filter 予測を用いた特徴点マッチングと mean-shift を組合せた粗密探索に基づく特徴点追跡. 画像電子学会誌, Vol. 43, No. 3, pp. 318–329, 2014.
- [19] 藏野隼二, 山本大樹, 林昌希, 片岡裕雄, 青木義満. スポーツ映像解析に向けたパス成功確率マップの作成. 映像情報メディア学会技術報告, Vol. 37, No. 36, pp. 45–48, 2013.
- [20] 望月宏祐. スマートフォン用の分光ベース cg レンダリング手法. デザイン学研究, Vol. 61, pp. 41–50, 2015.