

2017 年度 卒 業 論 文

2D キャラクターイラストによる 3D モーションデータ  
を用いたアニメーション補間手法に関する研究

指導教員：渡辺 大地 准教授

メディア学部 ゲームサイエンス プロジェクト

学籍番号 M0114155

栗原 一浩

2018 年 3 月

## 2017年度 卒業論文概要

論文題目

2D キャラクターイラストによる 3D モーションデータ  
を用いたアニメーション補間手法に関する研究

メディア学部

学籍番号：M0114155

氏名

栗原 一浩

指導  
教員

渡辺 大地 准教授

キーワード

奥行、イラスト、補間、画像変形、アフィン変換

近年 2D イラストを用いたアニメーションは需要が高く、2D イラストを用いたアニメーション技術、とくにセルアニメーションだけではなくモーフィングなどを用いて少ない枚数の原画でイラストを動かす技術が注目されるようになってきた。しかし、2D イラストを用いたアニメーション技術は動画としてだけではなく、インタラクティブに変化するデータを受け取り、リアルタイムに適用する技術もでてきている。2D イラストを入力データとし、3D として描画する技術が使用されている Live2DEuclid など存在し、アニメーション技術は近年著しい進化を遂げた。

これらの技術の殆どは顔や表情部分など頭に重点を置いており、体に対するアプローチは少ない。3D モデルを使えば体へのアプローチをした際の描画に対する課題は解決するものの、3D モデリングは 2D のイラストに比べ時間や技術に対する学習コストも多く、専門的知識のない人にとって複雑なモデルの制作は困難である。

入力したイラストに対し 3D で作成されたモーションデータの適用ができれば、様々な状況において即座に対応できる 2D キャラクターアニメーションの手法が確立できるという発想に基づき、2D イラストを用いたキャラクターの入力に、3D で作成したモーションデータの適用を実現することを目的とした。

その結果、本研究の目的である 2D イラストに 3D で作成されたモーションデータを用い、アニメーションを適用する手法の実装に成功した。問題点として、スケーリングが保たれず、特定の状況において意図しない描画が行われてしまうことを確認した。

# 目次

第1章	はじめに	1
1.1	研究背景と目的	1
1.2	論文構成	3
第2章	提案手法	4
2.1	キャラクター画像素材の用意	4
2.2	入力するデータ	6
2.3	実装アルゴリズムの概要	8
2.4	キャラクター画像素材のパーツ変形アルゴリズム	13
第3章	結果と考察	16
3.1	実行結果	16
3.2	考察	21
第4章	まとめ	22
	謝辞	23
	参考文献	24

# 目 次

1.1	モデルの差し替え例 . . . . .	3
2.1	使用するキャラクターのデザイン画像 . . . . .	5
2.2	図 2.1 を関節毎でパーツ分けしたもの . . . . .	5
2.3	キャラクターの画像素材を関節ごとでパーツ分けし、整列させたもの。 . . . . .	6
2.4	MMD 上で再生されているモーションの様子 . . . . .	7
2.5	実装ツール上で動いている様子 . . . . .	8
2.6	パーツの配置イメージ . . . . .	9
2.7	パーツの回転後配置イメージ . . . . .	10
2.8	キャラクター画像素材のパーツを仮に置いた様子（肩から肘にかけて腕パーツの例）	11
2.9	パーツの変形後配置イメージ . . . . .	12
2.10	キャラクター画像素材のパーツ各頂点とその名称 . . . . .	13
2.11	キャラクター画像素材のパーツ各頂点とその名称 . . . . .	14
2.12	キャラクター画像素材のパーツを画像変形をして置いた様子 . . . . .	15
3.1	「走る」アニメーションを再生したもの . . . . .	17
3.2	「走る」アニメーションにおいて 3 番の問題を拡大したもの . . . . .	17
3.3	「左手でパンチする」アニメーションを再生したもの . . . . .	18
3.4	「左手でパンチする」アニメーションにおいて 4 番の問題を拡大したもの . . . . .	18
3.5	「右手でパンチする」アニメーションを再生したもの . . . . .	19
3.6	「右手でパンチする」アニメーションにおいて 4 番の問題を拡大したもの . . . . .	20
3.7	「野球の投げるフォーム」アニメーションを再生したもの . . . . .	20

# 第 1 章

## はじめに

### 1.1 研究背景と目的

近年 2D イラストによるアニメーションの需要は増えてきており、少ない枚数の原画をモーフィングというある形から別の形へ滑らかに変化させる加工手法により、画像を変形させ動かす技術が近年注目されてきた。このようなモーフィングを用いたアニメーションが使われるようになった理由として、複数あるキーフレーム（コマ）の間を補間できるようになり、一枚一枚描かなくてよく、作業も効率化されたことが挙げられる。モーフィングを用いたアニメーションに関する研究も長利ら [1] や松延ら [2] などによって多く報告されている。しかしこれまでのモーフィングを用いた手法には課題があり、キャラクターにおいてアングルが変えられないなど、3 次元的な表現が困難である。激しい動きが苦手であるということが挙げられ、そしてこういった課題はシチュエーションに合わせ、原画を増やすことにより解決していた。これらの課題は 3D モデルを用いれば解決することができる。しかし、3D はモデルの作成は、2D のイラストよりも時間や技術に対する学習コストなども多く、専門的知識のない人にとって複雑なモデルの制作や表現は困難という問題がある。それらの課題を解決したツールとして Live2D[3] や SpriteStudio[4] などのツールが挙げられる。

特に Live2D の Euclid というバージョンからはこれまでのモーフィングを用いたアニメーション手法の課題を解決し、なおかつ 2D イラストを原画にしつつ立体表現を可能にし、ゲームや動画作品、その他アプリにも多く起用されている。

一例として、Live2D で作成されたデータを FaceRig[5] にインポートしコミュニケーションツールとして使用することができる物も出てきており、また、同様の技術を用いたツールを使い取り入れているものも、ゲームコンテンツ [6] [7] [8] において多く利用されるなど、少ない枚数の原画イラストによるアニメーションは非常に注目されている。そしてこういった 2D のイラストを原画とし、様々な動きや立体表現を実現する手法はこれまでも古澤ら [9] や北村ら [10] などにも報告され、モーフィングを用いた 2D キャラクターの表情を変化させるなどの手法も古泉ら [11] や白石ら [12] などによって多く報告されている。

しかし、これらのツールなどで作成されたものはほとんど顔を重点に充てており、2D イラストを用いて少ない原画でモーフィングや画像変形を用いた手法は体に対するアプローチが少ない。Live2DEuclid などは 2D と 3D を組み合わせたハイブリッド型にすることで体に対するアプローチをしており、体の部分は 3DCG ではあるがイラスト調や絵画調など、レンダリングをするというノンフォトリアリスティックレンダリング [13] と呼ばれている手法を用いられている。そしてこのような 2D のイラストのように見せる 3DCG なども伊藤ら [14] や奥屋ら [15] などによって多く研究報告されている。しかし、3D はモデルの作成やイラスト調に見せる手法は、2D のイラストよりも時間や技術に対する学習コストなども多く、専門的知識のない人にとって複雑なモデルの制作や描画表現は困難という問題がある。

そこで 2D イラストに 3D で作成されたモーションデータの適用ができれば、体部分全体における限定的なアニメーションしか実現できなかった課題を解決できる手法が確立できると仮説を立てた。その理由として、3D におけるキャラクターアニメーションは、主に 2 種のデータで構成されており、3D モデル（形状データ）とアニメーションを構成するモーションデータが存在し、

そして 3D モデル（形状データ）とモーションデータは相互に差し替えが可能であることである。

図 1.1 は同じモーションデータを用いてアニメーションを適用し、モデルを差し替えた例である。

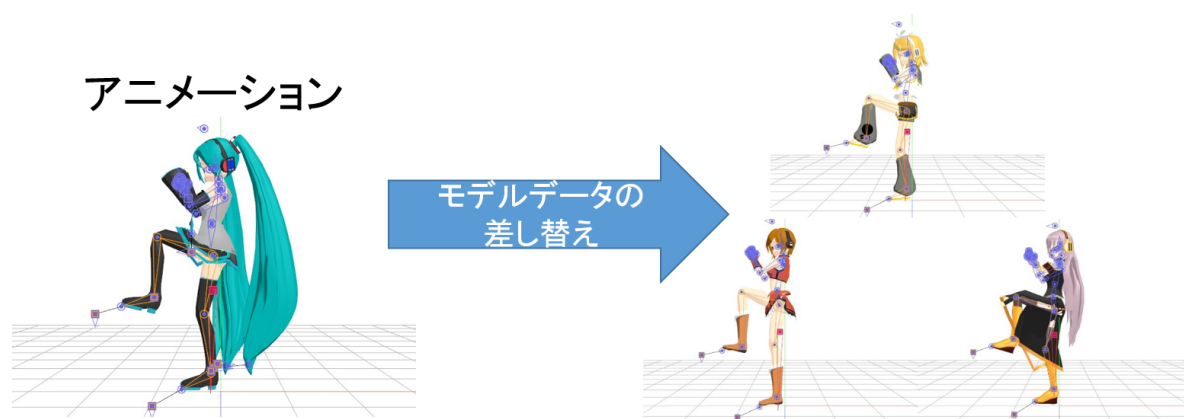


図 1.1 モデルの差し替え例

これらを踏まえ、モデル部分を 2D イラストを用いたキャラクターに差し替えることができれば 2D キャラクターにおいてモーションデータを適用できるのではないかと考えた。

そこで本研究では 2D イラストを用いたキャラクターの入力と、それに 3D で作成されたモーションデータを体へ適用する手法を提案し、実装することを目的とした。手法の概要としては、キャラクターの画像をキャラクターの関節部分ごとに分け、入力されるモーションデータに基づき単位時間で移動する点群データの位置ベクトルをスクリーン座標系へ投影した後、投影した点群データの位置ベクトルに合わせ関節ごとに分けたキャラクターの画像を配置していく。この手法により、2D イラストを用いたキャラクターへ 3D で作成されたモーションデータを体へ適用することができた。しかし、特定の条件下により意図しない描画を行うという問題を確認した。

## 1.2 論文構成

本論文は全 4 章にて構成する。構成は 2 章にて本論文の手法について述べ、3 章では本手法の検証について述べる。そして 4 章にてまとめを述べる。

## 第 2 章

# 提案手法

本章では本研究で提案する手法について述べる。2.1 節では用意するキャラクター画像素材について述べる。2.2 節では外部からのモーションデータの受け取り、本実装ツール上で適用させるまでについて述べる。2.3 節では実装するアルゴリズムの概要とイメージを述べる。2.4 節では関節座標にキャラクター素材を配置していく際の詳しいアルゴリズムについて述べる。

### 2.1 キャラクター画像素材の用意

キャラクター画像は図 2.1 のものを用意した。



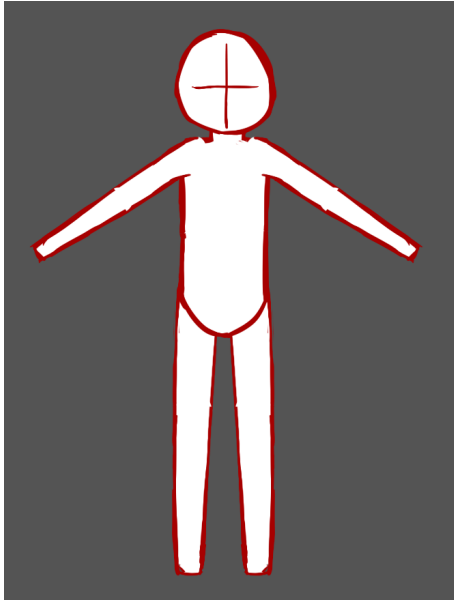


図 2.1 使用するキャラクターのデザイン画像

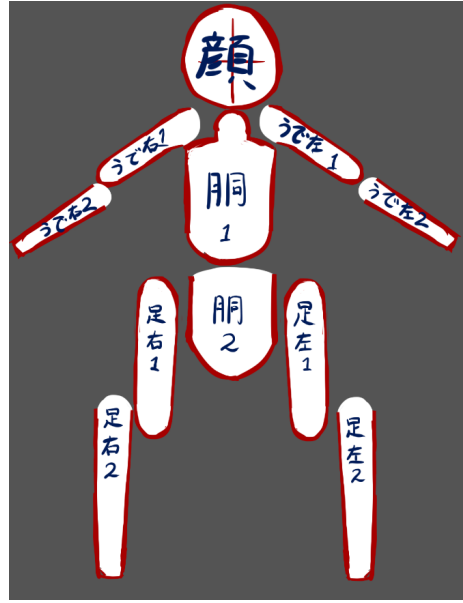


図 2.2 図 2.1 を関節毎でパーツ分けしたものの

次に、キャラクターイラストを関節ごとで区切る。そして各関節を分けたものを「パーツ」と呼称する。キャラクターイラストを各関節ごとで区切り、それぞれパーツ化したものが図 2.2 である。

次に、実装ツール上でのデータ管理の簡易化のため、先ほど関節で区切ったキャラクターイラストを整列し、配置する。配置した各パーツを 1 枚にまとめたものがキャラクター画像素材である。キャラクター画像素材におけるパーツの配置は、矩形でパーツ描画部分を切り出した時、他のパーツにおける描画領域が矩形で切り出した描画部分内に入らないように配置する。上側をキャラクターにおける関節の始点側、下側を関節の終点側になるように各パーツを配置することとする。なお、キャラクター画像素材はキャラクターのパーツを示す描画領域以外を透過部分とする。実装ツール上では、一枚のキャラクター画像素材から矩形で 1 つのパーツを切り出し描画する。

図 2.3 はキャラクター画像素材に各パーツをまとめ、整列させた様子である。

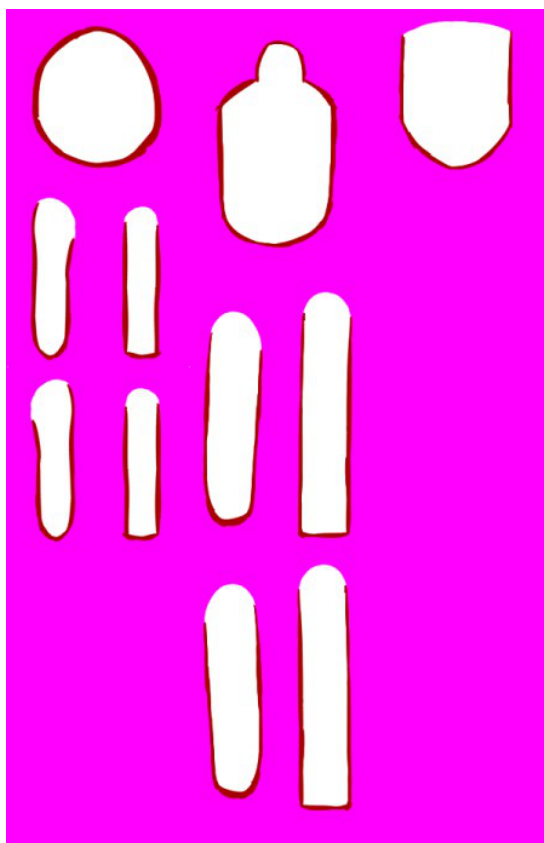


図 2.3 キャラクターの画像素材を関節ごとにパーツ分けし、整列させたもの。

## 2.2 入力するデータ

本研究では、実装ツール上で用いるモーションデータに入力する元データとして MMD[16] のモーションデータ（VMD モーションデータ）を採用した。MMD のモーションを採用した理由としては、モーションの種類が数多く容易に差し替えることができるからである。そして、本研究における実装部分に DX ライブラリ [17] を採用したのも MMD データの導入が容易にできることから採用した。図 2.4 は実際に MMD 上で再生し、MMD で用いられるボーンデータを表示したものである。

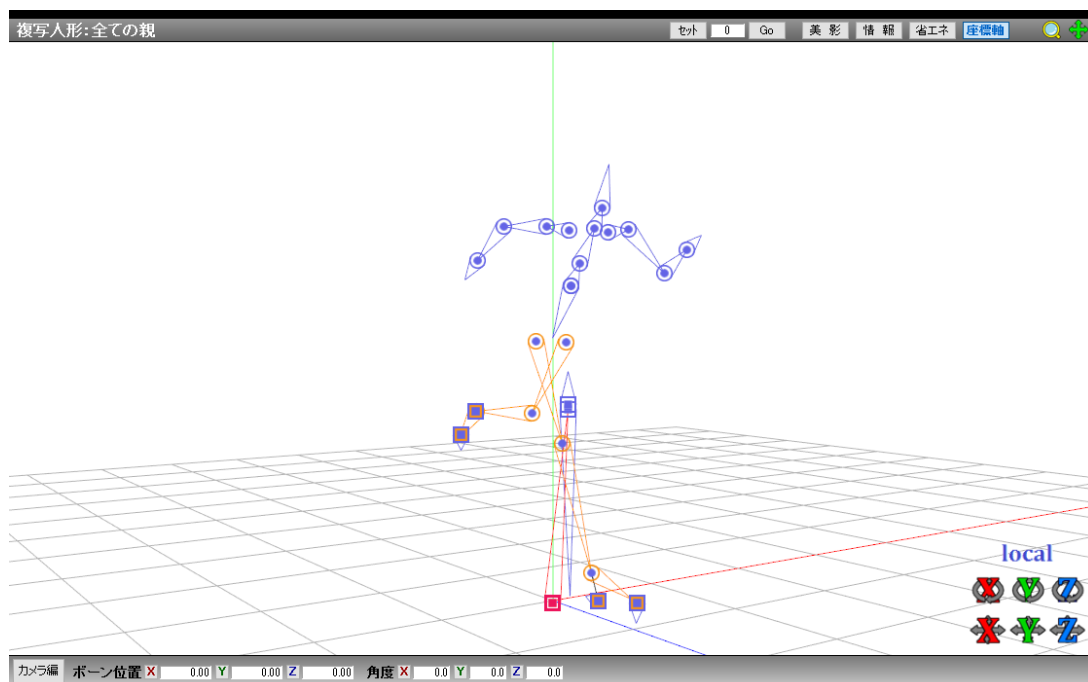


図 2.4 MMD 上で再生されているモーションの様子

本研究では実装ツール上で用いるデータとして定義している「関節座標」と「2D 関節座標」を用いて提案手法を実現した。関節座標とは、ワールド座標系における個別の関節の節を表す点群データの位置ベクトルのことを指す。この時、関節座標は MMD から DX ライブラリへ入力された情報を基に、単位時間により座標をモーションデータに従い移動させることでアニメーションを実現した。

2D 関節座標とは、関節座標をワールド座標系からスクリーン座標系に変換し、変換した 2 次元座標値のことである。ワールド座標からスクリーン座標への変換は、DX ライブラリの「ConvWorldPosToScreenPos」関数を用いて行った。ワールド座標を引数として入力するとスクリーン座標へ変換する機能をもつ関数である。図 2.5 は VMD モーションデータから DX ライブラリ上で入力を受け取り、実際にツール上で描画だけしたものである。

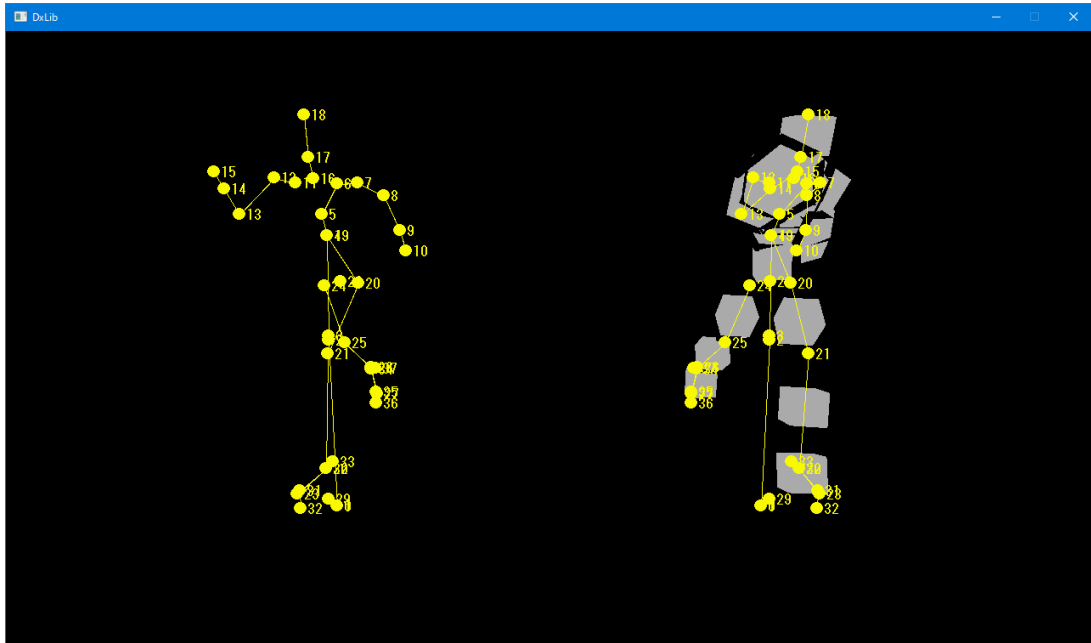


図 2.5 実装ツール上で動いている様子

なお、MMD から DX ライブラリへの入力を受け取る際は、DX ライブラリにて定義されている「MV1GetFramePosition」関数を用いた。MV1GetFramePosition 関数は入力された 3D モデルにおけるボーンデータの各頂点の座標を返す関数である。

## 2.3 実装アルゴリズムの概要

実装内容の概要としては、関節座標から一つの始点となる点の 2D 関節座標と、関節座標から終点となる点の 2D 関節座標の、2 点の 2D 関節座標を参照し、それに合わせてキャラクター画像素材のパーツを配置する。キャラクター画像素材のパーツをどの 2D 関節座標に配置するかは手動で決める。

配置方法として、まずキャラクター画像素材における、1 つのパーツの始点側の関節部分となる座標と、終点側の関節部分となる座標を手動で決める。本論文ではこれらの始点側と終点側の座標を、「パーツ関節座標」と呼称する。この時、パーツ関節座標における 2 点の識別は、パーツ関

節座標の始点側、パーツ関節座標の終点側、と明記する。

関節座標における始点部分の 2D 関節座標には、キャラクター画像素材のパーツにおける、スクリーン座標系上から見たパーツ関節座標の始点側に合わさるように配置する。図 2.6 は実際に配置した際のイメージを示したものである。

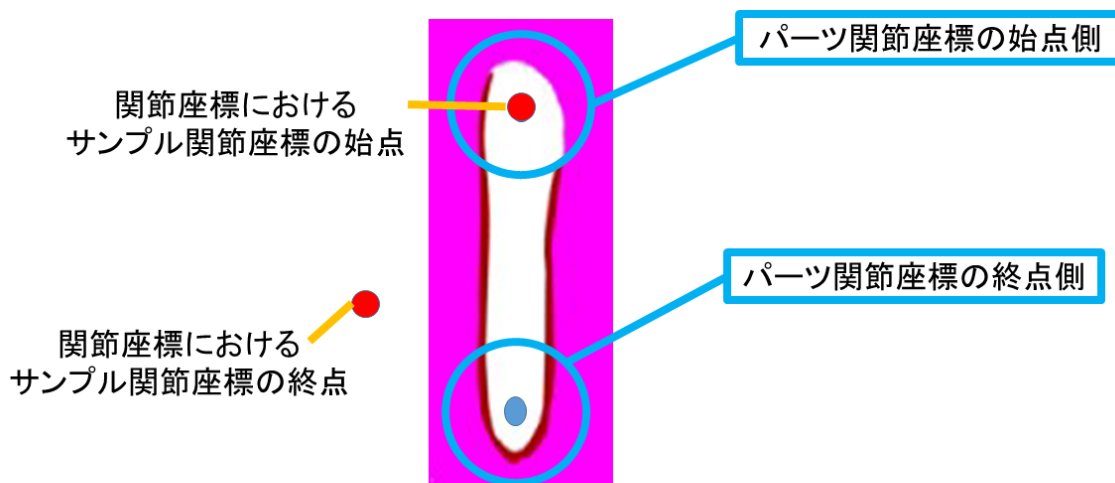


図 2.6 パーツの配置イメージ

キャラクター画像素材のパーツ変形方法は関節座標における、始点部分の 2D 関節座標から終点部分の 2D 関節座標へ引いた線分と、X 軸方向に平行な線分同士の角度を求め、キャラクター画像素材のパーツを求めた角度分回転する。図 2.7 は、回転した後の配置イメージである。

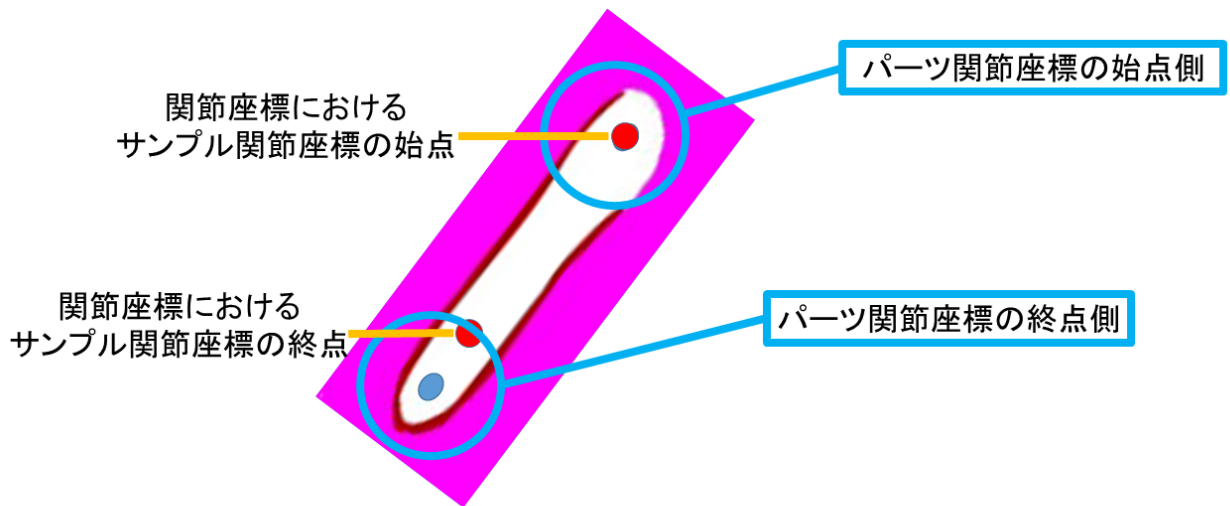


図 2.7 パーツの回転後配置イメージ

図 2.8 は実際に実装ツール上で配置した様子である。

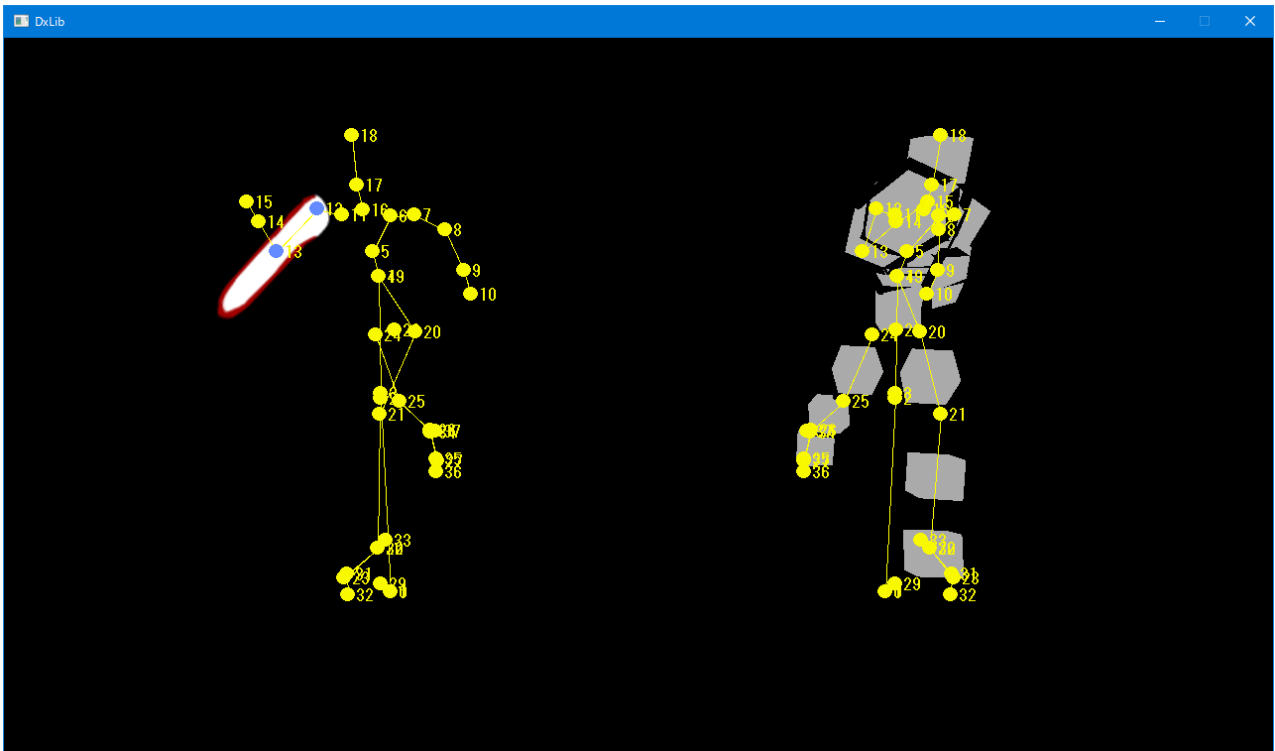


図 2.8 キャラクター画像素材のパーツを仮に置いた様子（肩から肘にかけての腕パーツの例）

ここで現時点で配置したパーツの縦の長さ、2点の2D関節座標の長さが異なるため、2点の2D関節座標の長さとパーツにおける、2点のパーツ関節座標が合うよう補正する。パーツにおけるパーツ関節座標の始点となる部分は関節座標の始点となる2D関節座標に配置し、パーツにおけるパーツ関節座標の終点となる部分は関節座標の終点となる2D関節座標に重ねるように配置する。

実際に変形させた後の配置イメージが図 2.9 である。

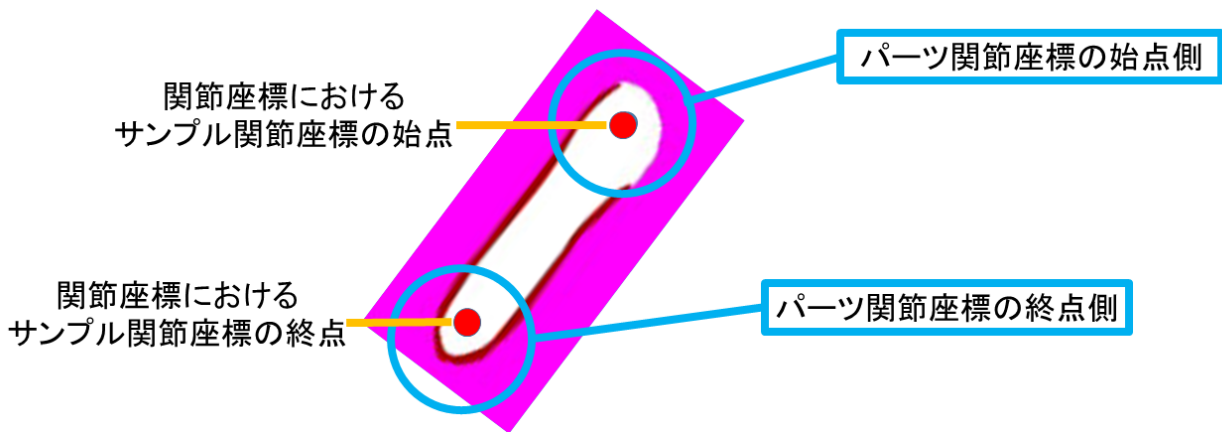


図 2.9 パーツの変形後配置イメージ

本手法ではキャラクター画像素材のパーツは三角形のポリゴンを二つ組み合わせ、四角形のポリゴンにしたものによって描画する。ポリゴンの各頂点はワールド座標からスクリーン座標に変換した後、2D グラフィックとして描画する。画像の変形には DX ライブラリにおける「DrawModiGraph」関数を利用している。キャラクター画像素材におけるパーツを描画する四角形の各頂点の座標を式によって移動することで実装する。キャラクター画像素材における個別のパーツ画像の左上となる頂点を「LeftTop」、右上となる頂点を「RightTop」、左下となる部分を「LeftBottom」、右下となる部分を「RightBottom」とする。図 2.10 はそのキャラクター画像素材のパーツ各頂点の座標を示した図である。



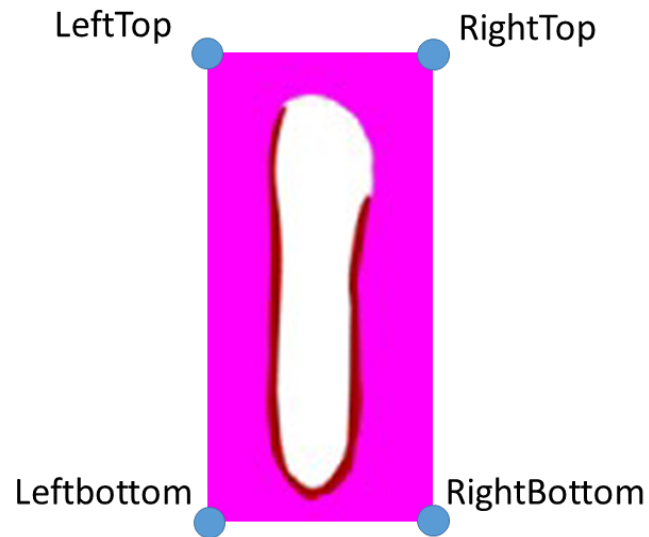


図 2.10 キャラクター画像素材のパーツ各頂点とその名称

## 2.4 キャラクター画像素材のパーツ変形アルゴリズム

キャラクター画像素材のパーツ変形アルゴリズムはアフィン変換を用いた。画像の変形においても野村ら [18] の研究論文からアフィン変換は有効であると判断したため、取り入れた。

キャラクター画像素材のパーツ各頂点の座標を次の式を用いて移動させることで実現している。

図 2.11 は式に使う要素を図示したものである。

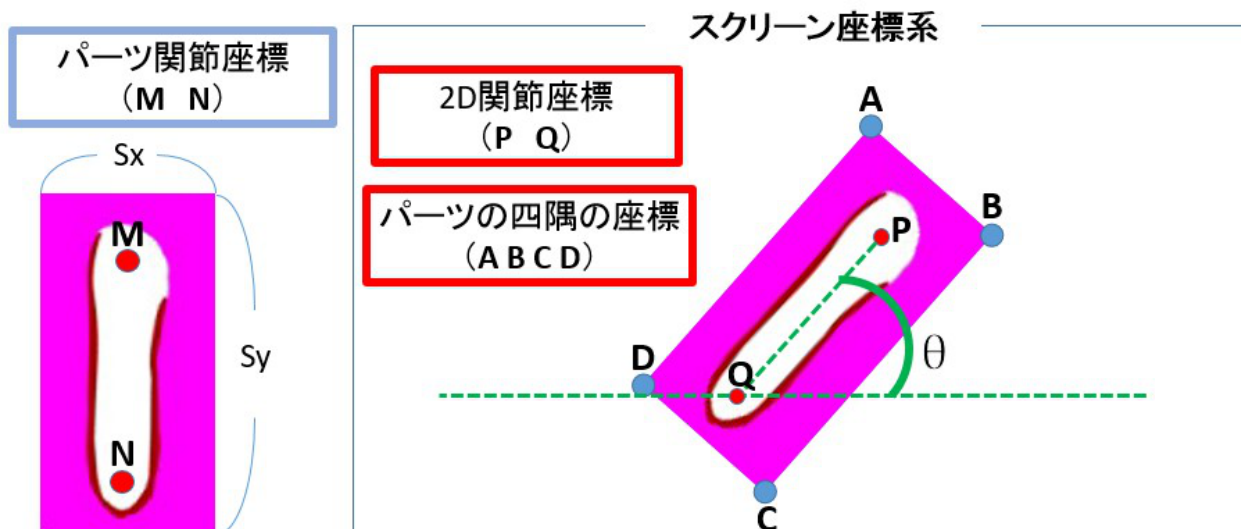


図 2.11 キャラクター画像素材のパーツ各頂点とその名称

パーツにおける LeftTop の変形後を **A**、RightTop の変形後を **B**、RightBottom の変形後を **C**、LeftBottom の変形後を **D** とする。キャラクター画像素材のパーツにおける画像の大きさ（単位はピクセル）を  $(S_x, S_y)$ 、キャラクター画像素材のパーツにおけるパーツ関節部分の座標（単位はピクセル、基準点  $(0,0)$  はキャラクター画像素材のパーツの左上とする）は、始点側の座標を **M**、終点側の座標を **N**、関節座標における始点部分の 2D 関節座標を **P**、関節座標における終点部分の 2D 関節座標を **Q**、**P** から **Q** へ引いた線分と、X 軸方向に平行な線分同士の角度を（ラジアン） $\theta$  とする時、**A**、**B**、**C**、**D** の座標はそれぞれ式 (2.1)(2.2)(2.3)(2.4) によって表すことができる。

$$\mathbf{A} = \begin{pmatrix} -M_x & -M_y & 1 \end{pmatrix} \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ P_x & P_y & 1 \end{pmatrix} \quad (2.1)$$

$$\mathbf{B} = \begin{pmatrix} S_x - M_x & -M_y & 1 \end{pmatrix} \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ P_x & P_y & 1 \end{pmatrix} \quad (2.2)$$

$$\mathbf{C} = \begin{pmatrix} S_x - N_x & S_y - N_y & 1 \end{pmatrix} \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ Q_x & Q_y & 1 \end{pmatrix} \quad (2.3)$$

$$\mathbf{D} = \begin{pmatrix} -N_x & S_y - N_y & 1 \end{pmatrix} \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ Q_x & Q_y & 1 \end{pmatrix} \quad (2.4)$$

式 (2.1)(2.2)(2.3)(2.4) を用いてキャラクター画像素材の腕部分のパーツを配置したのが図 2.12 である。

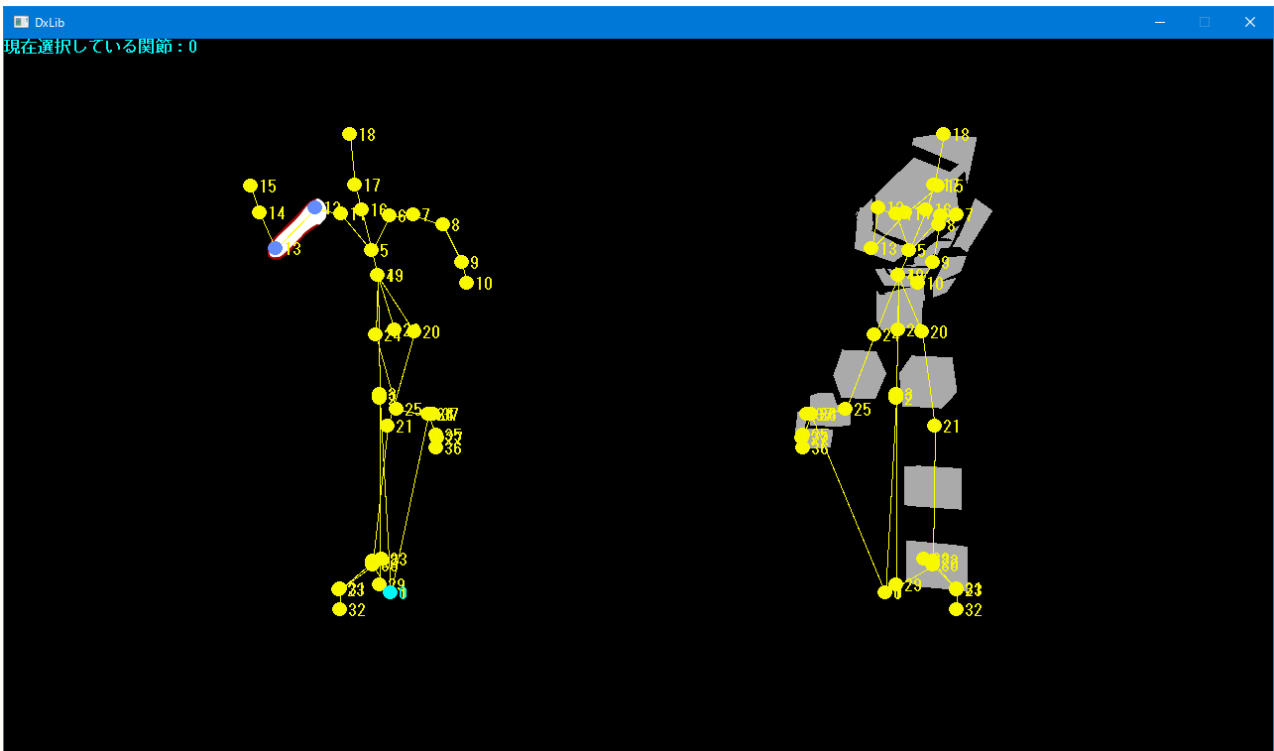


図 2.12 キャラクター画像素材のパーツを画像変形をして置いた様子

キャラクター画像素材において球体に近いパーツ（人間の頭など）は、式 (2.1)(2.2)(2.3)(2.4) を用いたアルゴリズムによる縦の長さを補正しない。なぜなら、その関節部分が潰れたり伸びたりしてしまい、意図した描画にならず破綻してしまうからである。

## 第 3 章

# 結果と考察

この章ではキャラクター画像素材のパーツに本手法を用いて実装した画像変形アルゴリズムを適用しアニメーションを再生した際に起きた結果や、問題の考察を述べる。

### 3.1 実行結果

図 3.1 は走りのモーションにキャラクター画像素材のパーツに対して画像変形アルゴリズムを適用し、描画したものである。左が本手法を用いて描画したモデル、右がモーションを確認できるように描画してある 3D モデルである。

さらにいくつかの違うモーションを適用し、キャラクターとしての問題がないか確認する。図 3.1,3.3,3.5,3.7 は実際にモーションを適用し、描画したものである。なお、ここで使用している MMD のモーションデータは個人で制作されたものが Web 上にて配布されているものである [19][20][21]。

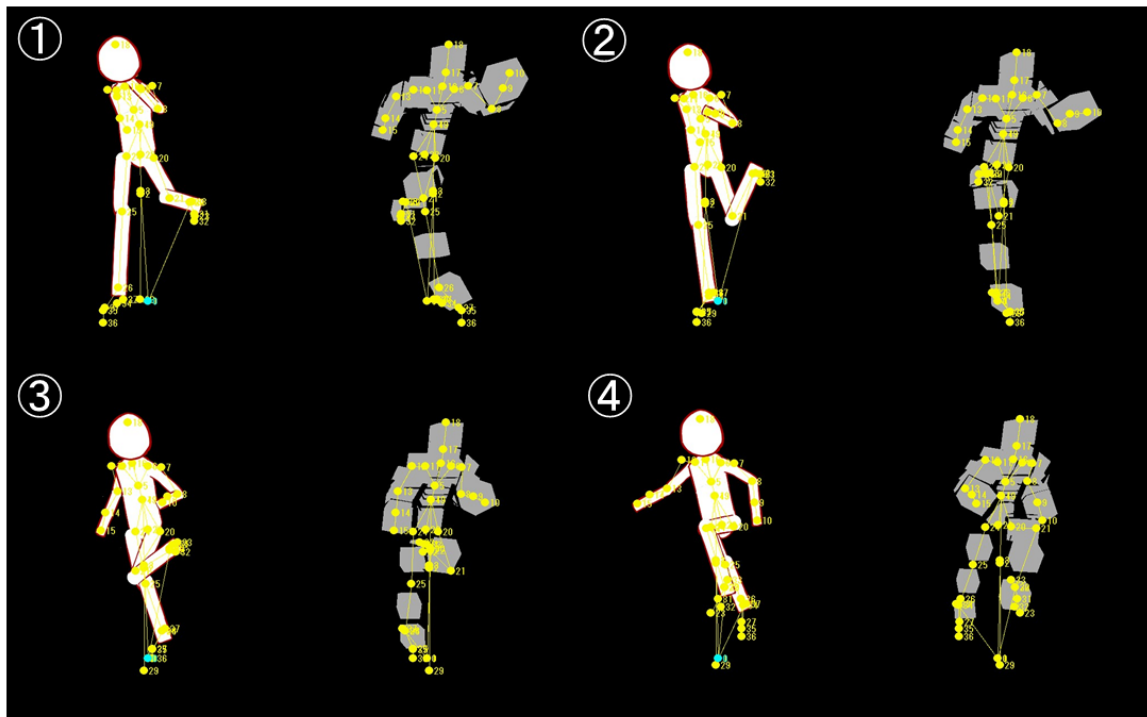


図 3.1 「走る」アニメーションを再生したもの

図 3.1 上では、関節部分はほぼ問題なくつながっているのが確認できる。しかし、図 3.1 上の 3 番の左足の膝から足先にかかるパーツは、足を曲げたら本来奥に描画されるが、手前に描画されてしまっている。図 3.2 は 3 番の左足に起こる問題を拡大したものである。

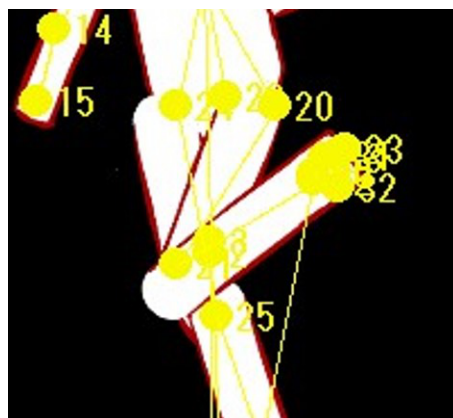


図 3.2 「走る」アニメーションにおいて 3 番の問題を拡大したもの

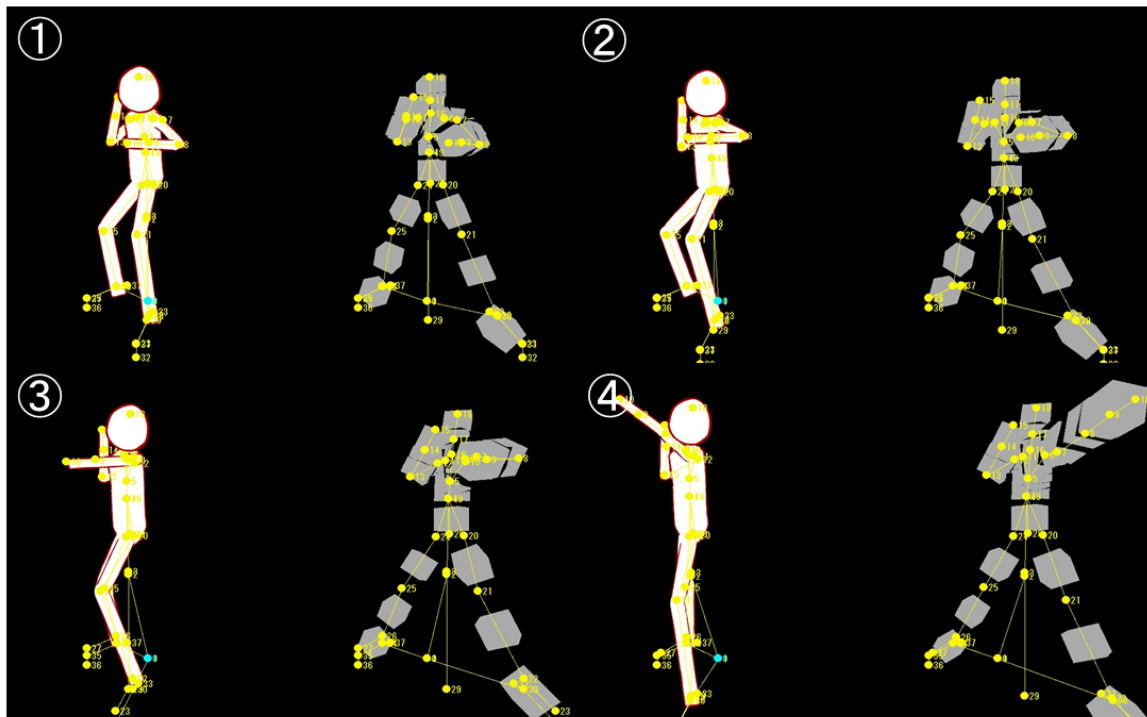


図 3.3 「左手でパンチする」アニメーションを再生したもの

図 3.3 上では 1～3 番までは関節部分、描画順共にほぼ問題なく理想的な描画結果が得られている。しかし、4 番において左腕より奥にある頭が手前に描画されている。図 3.4 は 4 番において左腕と頭の描画順の問題が起きている部分を拡大したものである。

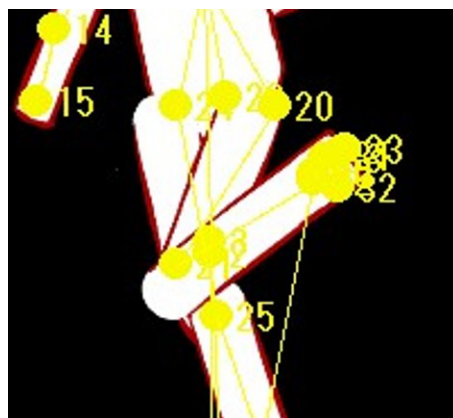


図 3.4 「左手でパンチする」アニメーションにおいて 4 番の問題を拡大したもの

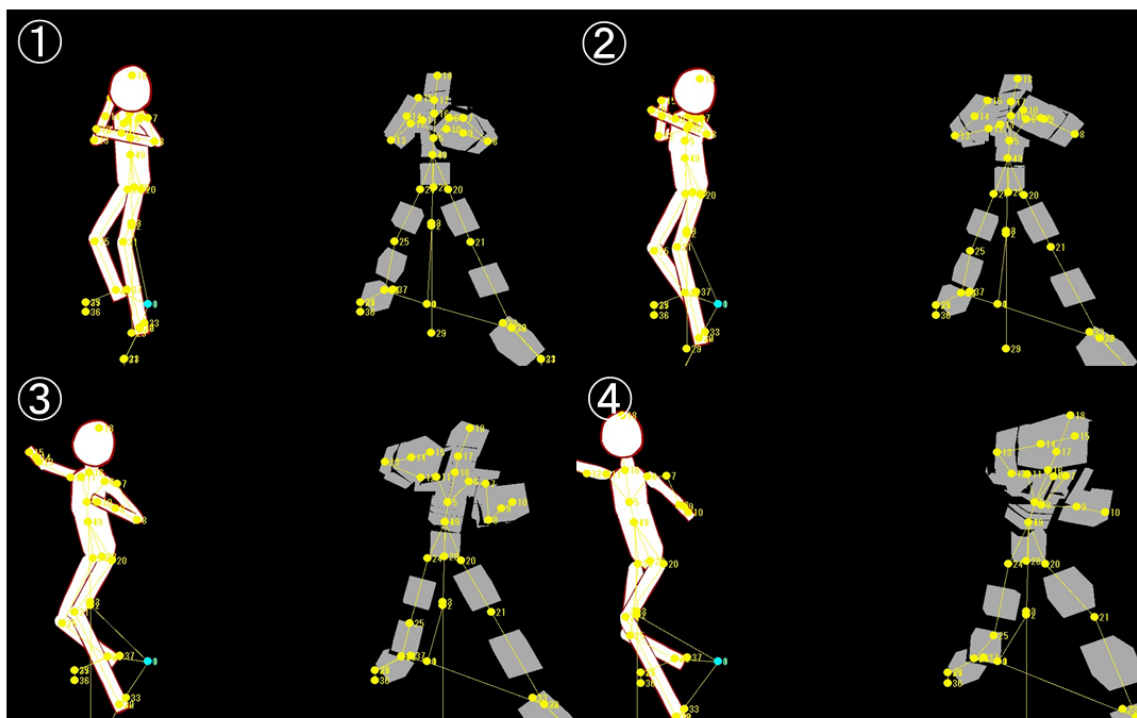


図 3.5 「右手でパンチする」アニメーションを再生したもの

図 3.5 上では 1~3 番までは関節部分、描画順共にほぼ問題なく、理想的な描画結果が得られている。しかし 4 番では、肘や膝の関節部分は問題が見られないものの、キャラクター画像素材のパーツが伸びてしまい、肩や足の付け根が胴体のパーツから離れている。

図 3.6 は 4 番で起きたパーツが伸びてしまい頭身が崩れてしまっている問題を、通常の頭身のものと比較したものである。

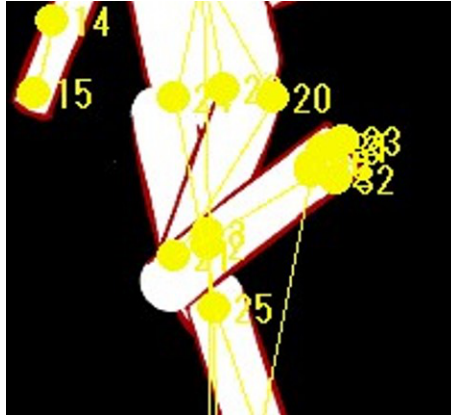


図 3.6 「右手でパンチする」アニメーションにおいて 4 番の問題を拡大したもの

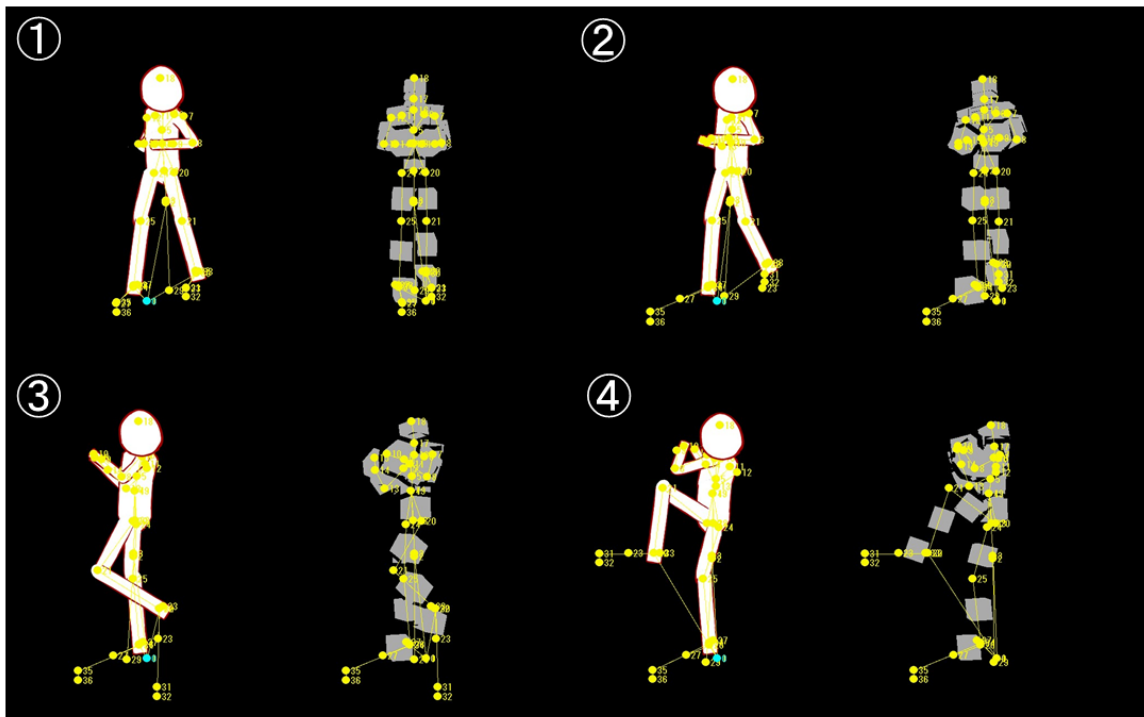


図 3.7 「野球の投げるフォーム」アニメーションを再生したもの

図 3.7 上では 1~4 番すべてにおいてほぼ問題なく理想的な描画が得られた。



## 3.2 考察

入力した MMD のモーションデータによって、キャラクターイラストとしてはまだ問題が見られたが、どれも 1 章で述べた関節がつながっていなかったりなどの破綻が解消できているのが確認できた。

見られた問題として、走るモーションによる足の描画順に対する問題と、左手でパンチするモーションによる腕と頭の描画順の問題に対しては、Z 方向に対する描画順の補正が出来ていなかったため、起こったと考えられる。

右手でパンチするモーションで起きた胴体から腕と足のパーツが離れてしまった問題は、全体的にボーンデータの関節座標がカメラに対して近くなった際に、Z 方向に対するスケールの補正をしていなかったのが原因だと考えられる。

## 第 4 章

### まとめ

この章では本論文のまとめを述べる。本研究では 2D のイラストによるキャラクターに 3D モーションデータを用いたアニメーションを適用させる手法を提案し、実装した。結果、本研究の目的である 2D イラストに 3D のモーションデータを適用させ、アニメーションさせることに成功した。

しかし問題点として、パーツがカメラに近づいた際のパーツのスケールが保たれない、パーツが意図しない部分から描画される、本手法を適用できるキャラクターのデザインが限定的などが挙げられる。

今後の展望として、奥行きを考慮したスケーリングの補間を実装すること、パーツの描画順に対するアプローチを実装の実現を挙げる。これらの実現ができれば本手法を用いて実装したツールのさらなる発展が旨せると考えた。

# 謝辞

本論文を執筆するにあたり、ご指導頂きました渡辺先生、阿部先生に心より感謝いたします。

また、様々な相談に親身に応じてくださった研究室のメンバー、友人たちにこの場を借りて深く感謝いたします。

## 参考文献

- [1] 長利健治, 齋藤豪. 異なる二つのイラスト間の補間によるアニメーション. 第 73 回全国大会講演論文集, 第 2011 巻, pp. 93–94, mar 2011.
- [2] 松延徹, 長原一, 岩井儀雄, 谷内田正彦, 鈴木俊哉. モーフィングによる高解像度高フレームレート動画の生成. 電子情報通信学会論文誌 D, Vol. J90-D, No. 4, pp. 1073–1084, 2007.
- [3] Live2D. 株式会社 Live2D. <http://www.live2d.com/ja/>. 参照:2017.12.10.
- [4] SpriteStudio. OPTPiX SpriteStudio — ウェブテクノロジー. <http://www.live2d.com/ja/>. 参照:2017.12.10.
- [5] FaceRig. FaceRig. <https://facerig.com/>. 参照:2017.12.13.
- [6] グランブルーファンタジー. グランブルーファンタジー. <http://granbluefantasy.jp/>. 参照:2018.1.19.
- [7] マギア・レコード. 「マギアレコード 魔法少女まどか☆マギカ外伝」公式サイト. <http://magireco.com/>. 参照:2018.1.19.
- [8] ファイアーエムブレム if. ファイアーエムブレム if — ニンテンドー 3DS — 任天堂. <https://www.nintendo.co.jp/3ds/bfwj/>. 参照:2017.12.13.
- [9] 古澤知英, 福里司, 岡田成美, 平井辰典, 森島繁生. 正面および側面のイラストからのキャラク

- タ顔回転シーンの自動生成. 情報処理学会研究報告. グラフィクスと CAD 研究会報告, Vol. 2014, No. 8, pp. 1–6, sep 2014.
- [10] 北村真紀, 金森由博, 鶴野玲治. 異なる視点から描かれたイラストからの 2.5d モデル生成. Technical report, 2014.
- [11] 古泉大輔, 橋本康弘, 陳ユ, 大橋弘忠. 感性データ学習による顔イラストへの表情付け. 横幹連合コンファレンス予稿集, Vol. 2007, pp. 135–135, 2007.
- [12] 白石路雄, 山口泰. 絵画風画像モーフィング. 情報処理学会論文誌, Vol. 45, No. 2, pp. 659–667, feb 2004.
- [13] 斎藤隆文. ノンフォトリアリスティック・レンダリング. 日本印刷学会誌, Vol. 51, No. 4, pp. 262–267, 2014.
- [14] 伊藤翔愛, 藤代一成, 大野義夫. ポリゴンモデルのイラスト風半透明表現. 第 73 回全国大会講演論文集, 第 2011 巻, pp. 79–80, mar 2011.
- [15] 奥屋武志, 坂井滋和. リアルタイムレンダリングにおける投影面上での曲がり具合を考慮した輪郭線描画 (画像処理・感性, 映像表現・芸術科学フォーラム 2016). 映像情報メディア学会技術報告, Vol. 40.11, pp. 29–32, 2016.
- [16] MMD. VPVP. <http://www.geocities.jp/higuchuu4/>. 参照:2018.1.19.
- [17] DX ライブラリ. DX ライブラリ置き場 HOME. <http://dxlib.o.oo7.jp/>. 参照:2017.12.10.
- [18] 野村由司彦, 原田裕次郎, 藤井省三. 画素データレベルでのアフィン変換画像のマッチング. 電子情報通信学会論文誌 D, Vol. Vol.J75-D2, No. 9, pp. 1498–1503, 1992.
- [19] 走りモーション. 【MMD】アニメ走りっぽいモーション配布 - ニコニコ動画. <http://www.nicovideo.jp/watch/sm22739600>. 参照:2017.11.27.
- [20] フリックージャブモーション. 【MMD】フリックージャブモーション【配布】 - ニコニコ動

画. <http://www.nicovideo.jp/watch/sm25366732>. 参照:2018.1.8.

[21] 投球モーション. 【第 11 回 MMD 杯本選】 こんにちは、ミク・ギターです - ニコニコ動画

. <http://www.nicovideo.jp/watch/sm21609802>. 参照:2018.1.8.