# アニメーション技術

1. 技術紹介と解説

# はじめに

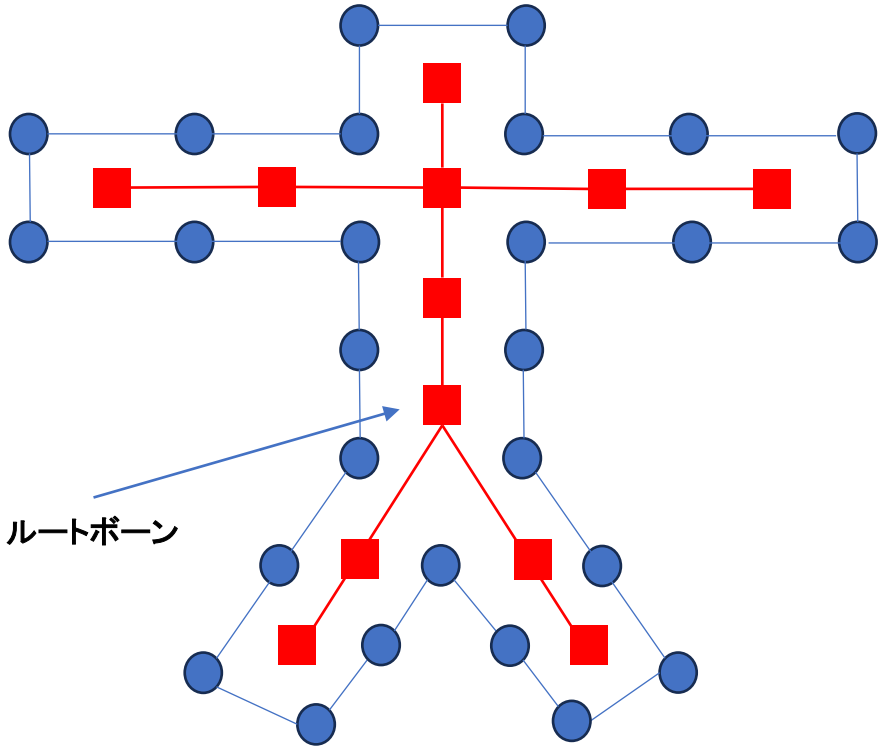
言い訳ではありませんが、私の専門がアニメーションではないので、
実装等の、とても細かい所までは詳細には語りません。ご了承ください。

# アニメーション技術 もくじ

# スケルタルアニメーション

1. アニメーションの基礎！

# スケルタルアニメーション（ざっくり解説１）




キャラクターはスケルタルと呼ばれる、

階層型の骨格（ボーン）を保持しています。

※左の赤い部分

スケルタルアニメーションは、

スケルタルが指定された姿勢へと動作していく時、

各頂点が、階層構造に基づいて動かされていくことで、

アニメーションを実現しています。

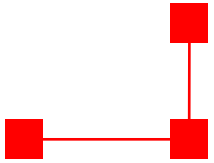
「なんのこっちゃ？」ってなると思うので詳しく解説します。

# スケルタルアニメーション（ざっくり解説２）

ボーン

ボーン2つ

動かしてみた



ボーンとは、二つの頂点を持つものです。

正に骨のような形で、これが変形することはありません。

このボーンが複数繋がっていくことで、

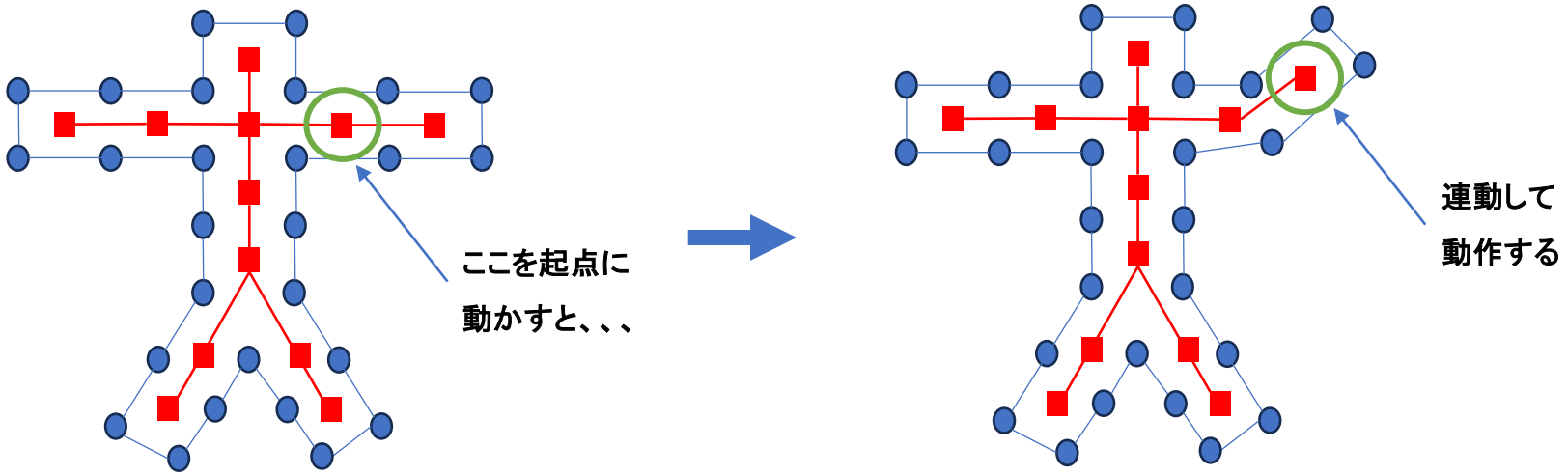
先ほどのような、人型とかの形を作ります。

そして、このボーンが回転など動作することで、

実際にアニメーションを実現することが出来ます。

※ボーン回転はちょっと面倒な計算をする必要があります。

# スケルタルアニメーション（ざっくり解説 3）




骨格（ボーン）は、階層構造で管理されているため、

親となる行列に変化があると、対応した子ボーンにも影響を与え動作します。

※子ボーンの行列の変化は、親ボーンに影響を与えない。

# スケルタルアニメーション（ざっくり解説 4）

また、スケルタルアニメーションは、滑らかな頂点の動きを実現するために、
各ボーンの影響度（ウェイト値）を指定する場合がほとんどです。

ざっくりとした内容ですが、アニメーションを学びたいなら、
ぜひ一度ちゃんと調べて見てください。

良資料：

ゲームプログラマーを目指すひと スキンメッシュアニメーションの解説

その27 アニメーションの根っこ：スキンメッシュアニメーション（ボーン操作）

その61 完全ホワイトボックスなスキンメッシュアニメーションの解説

# モーションブレンド

1. 滑らかなアニメーション遷移に向けて

# モーションブレンド（ざっくり解説 １）

モーションブレンドは、複数のアニメーションをいくつかのブレンドパラメーターを使って、
滑らかにブレンド出来るようにするためのモノです。

例えば、
立っているアニメーションから歩いてるアニメーション、走っているアニメーションへと、
変化していく流れを補間しながら遷移してくれます。

他の例では、
足は走っているけど、上半身は攻撃しているなど、複雑なアニメーションを作ることも可能です。

# モーションブレンド（ざっくり解説 2）

基本的な実装としては、

「歩いてる：100%　走っている：0%」から徐々に割合を変えて、

「歩いてる：0%　走っている：100%」へと変化させていくクロスフェードだと思います。

しかし、アニメーション遷移が完了する前に、

他のアニメーションが開始すると破綻してしまう可能性があるなどいくつか問題もありました。

そこで、**慣性補間によるアニメーション遷移**が「FF7R」などで採用されているので、

実装するならそちらをやってみてはいかがでしょうか？

※アニメーション遷移が、破綻しにくいだけなので可能性はあると思います。

# モーションブレンド（ざっくり解説 3）

**良資料:**

慣性ベースドなアニメーションブレンド（解説編） - ほげたつブログ (hatenablog.com)

慣性ベースドなアニメーションブレンド（実装編） - ほげたつブログ (hatenablog.com)

# IK（インバースキネマティクス）

1. 物理の力でゴリ押そう！

# IK（ざっくり解説 1）

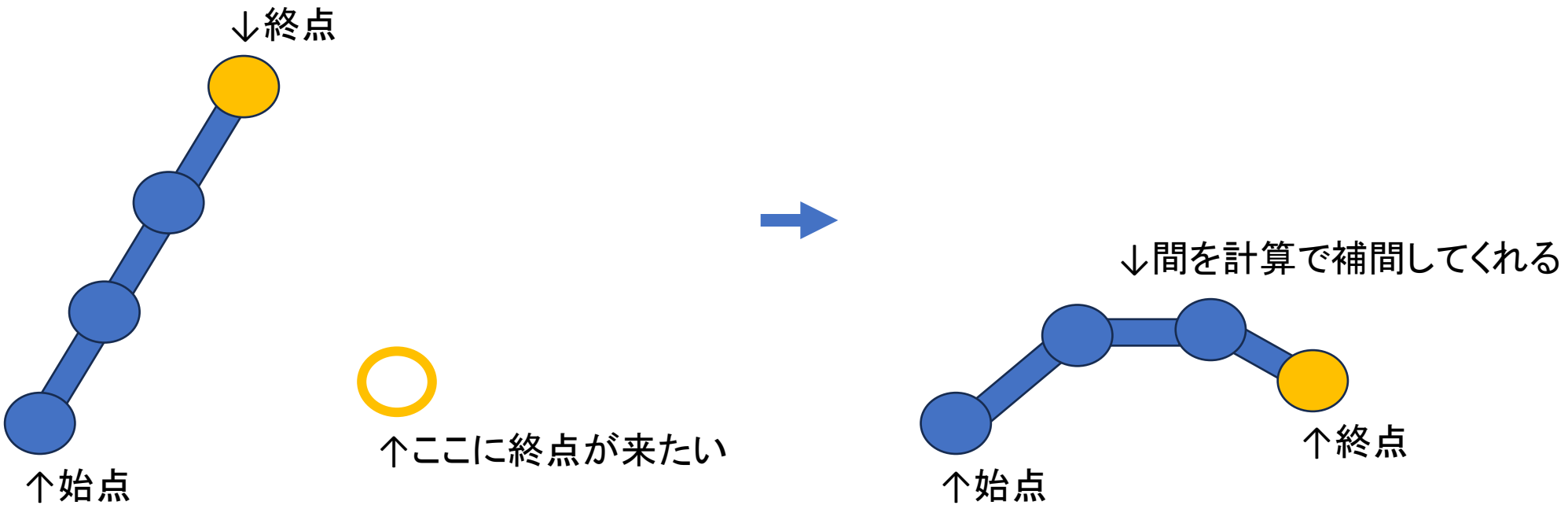
別名は「逆運動学」とも呼ばれ、その名の通り、

ボーンの始点と終点、姿勢が決まると、

逆算して、その間を上手く繋ぐ関節角を求めることが出来る技術です。

これも「なんのこっちゃ？」と思うかもしれないので、もうちょっと解説します。

# IK（ざっくり解説２）




**始点と終点、姿勢が決まっていれば、その間の角度を求め、上手く繋いでくれます。**

**基本的には、ボーンが曲がる角度を設定することで、不自然な動作を防ぎます。**

# IK（ざっくり解説 3）

IKには、沢山の種類がありますが、そのうちいくつかを紹介します。

# IK（ざっくり解説 4）

**TwoBoneIK**

人間の腕などを表現する際に有効で、

手（ターゲット）と肘（ヒント）によって制御することが出来ます。

下記のFootIKの一種でもあります。

**FootIK**

キャラクターの足首の位置を制御することで、足の動きを制御します。

地面にめり込まないようにするなど、自然な動作にすることが出来ます。

FootIKには、

Two Bone IK、Spline IK、Hinge IK、Spring IK、Foot Roll IKなどの手法があります。

# IK（ざっくり解説５）

**Fullbody-IK**

**キャラクター全身の動作を制御する方法の一つです。**

**全身の動作を滑らかにすることが出来、よりリアルな動きになります。**

**しかし、高負荷であるため、Two Bone IKで代用されることが多いです。**

**Fullbody-IKには、**

**Jacobian IK、CCD-IK、FABR IKなどいくつかの手法があります。**

# IK（ざっくり解説 6）

IKは割と成熟した技術なので、わざわざ実装する必要はあんまりないかな？
っていうのはありますが、知っておいて損はないので調べてみる価値はあります。

**良資料**:

Inverse Kinematics(IK)について - SEGA TECH Blog

モーション編その③　IKを適用しよう！ | ぶち (note.com)

Three.js でIKを実装してみた - CodeLabo

# モーションマッチング

1. 数の暴力マッチングシステム

# モーションマッチング（ざっくり解説 1）

キャラクターを自然に動作させるために、アニメーションをステートマシンで作成した場合、
モーションが増える度に、ステートマシンが大きくなり、組み込みコストや修正が複雑化していきます。

そんな問題を解決しようと考えられたのが、このモーションマッチングという技術です。

# モーションマッチング（ざっくり解説 2）

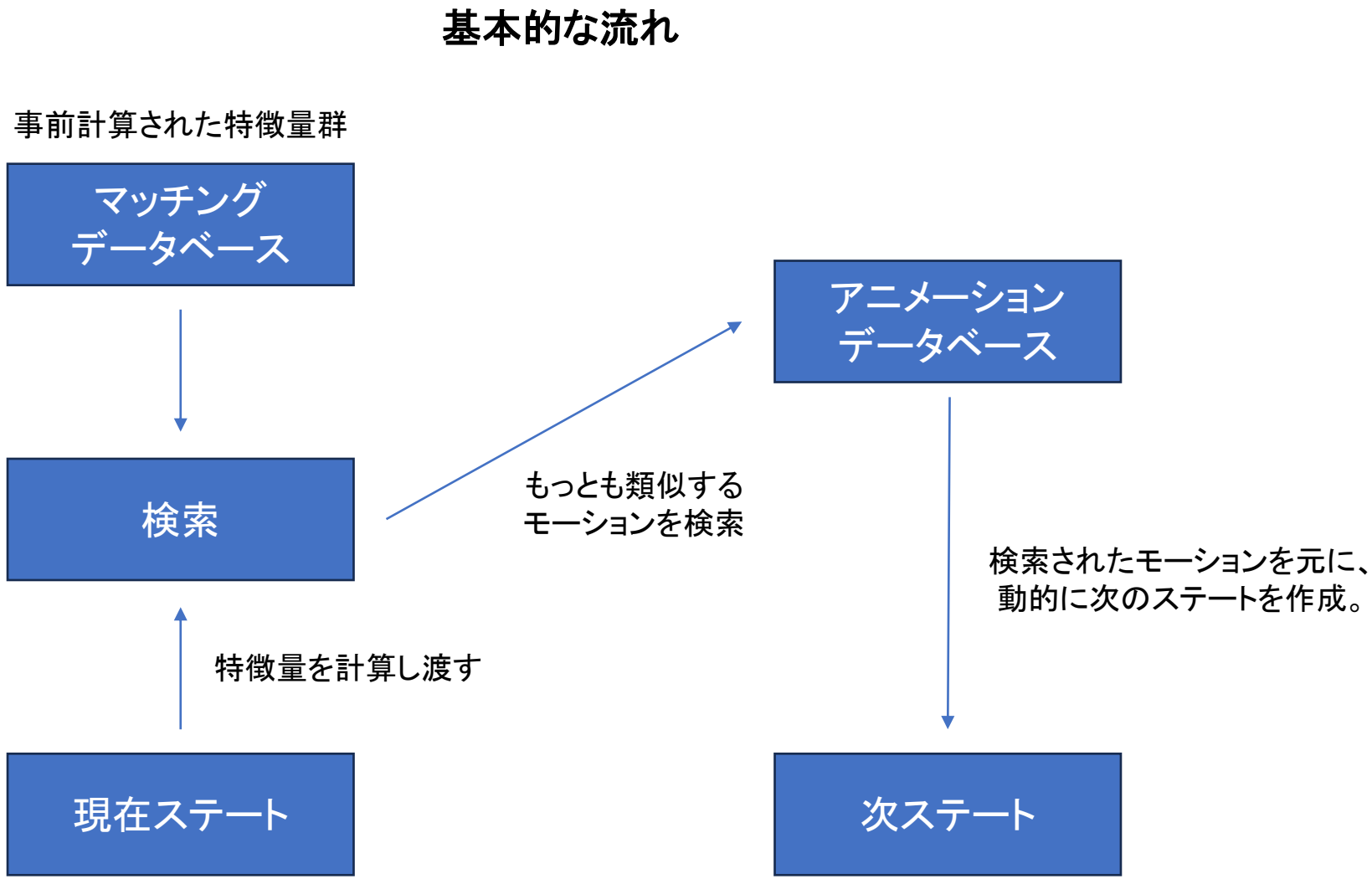
モーションマッチングとは、

予め用意したアニメーションデータとユーザーの入力を元に、

リアルタイムにモーションの生成を行う、データ駆動型のアニメーション技術です。

「リアルタイムにモーションを作成出来るの？」と思うとかもしれないので、少しだけ解説します。

# モーションマッチング（ざっくり解説 3）



# モーションマッチング（ざっくり解説 4）

先ほどの図の解説をします。

モーションマッチングでは、再生中のアニメーションよりも適したアニメーションを探し続けます。
その際、特徴量と呼ばれる基準値を元に比較していきます。

特徴量とは、関節位置や回転角度が簡単なデータとして考えれますが、
インタラクティブな要素を含む（ゲームの）場合、未来の動作軌道や速度を使用する場合があります。

そして求められた特徴量を元に、事前計算されたマッチングデータベースから類似する最適な
モーションを検索し、新しいステートとして動的なモーションの作成を行います。

# モーションマッチング（ざっくり解説５）

しかしモーションマッチングは、動的なモーションを作成するための検索コストが非常に高価です。

なぜなら、全フレームで検索をかけるため、
データベースの大きさに比例して処理負荷がかかるためです。
そこで、いくつか最適化をする必要があります。

最適化の方法はいくつかありますが、
一つはデータベース内で二つの大きさのAABBを使用し、検索数を減らすというものがあります。
もう一つはKD-Treeを使用するという方法です。
※どちらも当たり判定とかで使用される技術ですが、モーションマッチングでも有効です。

# モーションマッチング（ざっくり解説 6）

めちゃくちゃ優秀そうなモーションマッチングですが、

未来の軌道や速度といった特徴量の設定を正しく行わないと、

意図しないアニメーションが再生するなど問題もいくつかあります。

※この次の項目（進化版）で解説します。

**良資料：**

モーションマッチングのつくりかた (cesa.or.jp)

Motion Matchingによるキャラクター操作 (zenn.dev)

# モーションマッチング進化版

1. 深層学習を使った最適化

# モーションマッチング進化版（ざっくり解説 1）

先ほどのざっくりとした、解説の通り、
モーションマッチングにはいくつか問題点がありました。

それは、検索コスト、メモリ使用量、
特徴量の設定により意図しないアニメーションを再生するなどです。

今回紹介する進化版は、
その検索コストと、メモリ使用量、この二つを解決するというモノです。

# モーションマッチング進化版（ざっくり解説２）

ここから更に詳しく解説したいのですが、概要だけにさせていただきます。

※この進化版は僕はちゃんと調べた事がないのです。（本当にアニメーション専門の人じゃないので）

この進化版は「Learned Motion Matching」と呼ばれるものです。

内部は3っつのニューラルネットワークから構成されており、

元々あった、アニメーションデータベースが無くなっています。

この深層学習を導入したことにより、メモリ使用量を70倍ほど軽量化したそうです。

# モーションマッチング進化版（ざっくり解説３）

**良資料**:

Learned motion matching | ACM Transactions on Graphics

orangeduck/Motion-Matching: Learned Motion Matching example implementation and source code for the article "Code vs Data Driven Displacement" (github.com)

# リップシンクアニメーション

1. 音声とリンクする口のアニメーション

# リップシンクアニメーション

リップシンクとは、

セリフ（音声）に合わせて、モデルの口を動かすことを言います。

実装方法については、現在ほとんどのケースでAIが使用されています。

そのため、良資料を参考にして欲しいです。

※無理やり作るなら、音声の波形データから母音の数値に近いものを喋らせるようにすれば良い。

良資料：

機械学習によるリップシンクアニメーション自動生成技術とFINAL FANTASY VII REMAKE のアセットを訓練データとした実装実例 (cesa.or.jp)