

座標系の変換（MediaPipe → モデル空間）

MediaPipeの「ワールド座標」では、原点が左右の股関節の中心にあり、Y軸が上方向、Z軸はカメラに向かう前方向として定義されています¹。一方、glTFモデル（OpenGL基準）の座標系は右手系で、+Yが上、+Zがモデルの正面（前方）です²。したがって、MediaPipeの座標からモデル空間へ変換するには軸の入れ替えと反転が必要です。具体的には、たとえばMediaPipeの出力する座標\$(x, y, z)\$をモデル座標系に合わせて\$(x', y', z')\$に変換するには、**前後軸を反転**する操作が必要になります。これは、人物がカメラに正面を向いているMediaPipeの+Z方向を、モデルが正面を向く+Z方向に合わせ直すためです。最も簡単には、Y軸を共有しつつ、Z軸を反転する**180°のヨー回転**（Y軸周りの半回転）を適用します。例えば：

- $x' = -x$ （必要なら左右軸も反転して齟齬を修正）
- $y' = y$
- $z' = -z$

このように軸を入れ替えたり符号を反転させることで、MediaPipeのカメラ基準の座標をモデルのグローバル座標系（右手系、Yアップ、Z前方）にマッピングします¹²。変換後のポイント群はモデル空間上のワールド座標となり、以降のボーン計算に利用できます。

なお、**単一のカメラ映像に対するMediaPipeの推定**では左右が鏡映される可能性があります。上記のように軸変換することで、たとえば「右手」「左手」といった左右の骨もモデル上で正しく対応します。以降では、この変換後のモデル空間における関節点（33箇所）を基に計算を行います。

ボーンの方角ベクトル算出とワールド座標系での基準軸

次に、MediaPipeから得た各関節の位置データから、モデルの各ボーンが指す方向（**ボーン軸方向**）と、ボーンの握じれを決める**参照軸方向**（いわゆる前方方向）を算出します。ここではglTF Humanoid仕様の17ボーン（肩を含まない左右対称の骨格）について、それぞれ**ワールド空間での+Y軸方向**（ボーンの軸）および**+Z軸方向**（ボーンの前方向基準）を定義します。各ボーンのローカル座標系は、**+Y軸がボーン方向（親関節から子関節への方向）、+Z軸がそのボーンの前方向（解剖学的な前向き）**になるよう設定します。この約束に従って、MediaPipeの関節座標から以下の手順で各ボーンの基準軸ベクトルを計算します。

一般的な手順：親関節と子関節の3次元位置 \mathbf{p}_{parent} 、 \mathbf{p} は単に両者を結ぶベクトルです： \mathbf{d} が与えられれば、ボーン軸方向ベクトル \mathbf{d}

- ボーン軸方向（ローカル+Y軸に対応）
$$\mathbf{d} = \mathbf{p}_{child} - \mathbf{p}$$

これを正規化した $\hat{\mathbf{d}}$ がボーンの前方向ベクトルになります³。このベクトルはボーンの世界空間における長さ方向（親→子）を表します。

しかし、**ボーンの回転**を一意に定めるには軸方向だけでは不十分で、軸周りのねじれ（ツイスト）を決めるためにもう一つの参照ベクトルが必要です。そこで各ボーンについて、「前方」または「上方向」といった**第2の基準**

方向（ローカル+Z軸に相当）を設定します。この基準として、ボーンが属する関節平面の法線や、手足の場合は肢の曲げる方向、体幹の場合は身体の正面方向などを用います。以下、ボーンの種類ごとに詳細を説明します。

末端ボーン（四肢）の前方・上方向とロール推定

上腕・前腕・手、大腿（腿）・下腿（脛）・足といった末端のボーンでは、それぞれの関節の配置から**肢の平面**（曲げる方向の平面）を定義し、その法線ベクトルを使ってボーンのローカル+Z軸（前方方向）を決めます。併せて、この平面からボーンのひねり成分（ロール）を推定します。

- ・**上腕（肩～肘）**：上腕骨の軸 $\hat{\mathbf{d}}_{upperArm}$ は肩関節点（今回は肩ボーン省略のため胸部の肩位置に相当）から肘へのベクトルです。上腕の曲げ平面は、上腕軸と前腕軸で定まります。肘関節での曲げ方向を求めるために、肘→手首のベクトル $\mathbf{v}_{forearm}$ を用います。平面法線 $\mathbf{n}_{upperArm}$ を

$$\mathbf{n}_{upperArm} = \hat{\mathbf{d}}_{upperArm} \times \mathbf{v}_{forearm}$$

と定義し（肩→肘と肘→手首の外積）、これを正規化して上腕の**+Z方向**ベクトルとします。この $\hat{\mathbf{n}}_{upperArm}$ は上腕骨の「前方」を表し、肘を伸ばしたときに肘が向く方向（つまり肘関節の曲がる方向）に対応します⁴。例えばTポーズで腕を真横に伸ばし手のひらを下に向けた場合、上腕の曲げ平面はほぼ垂直で、 \mathbf{n} で算出）。 \mathbf{n} は身体の前方向（グローバル+Z）に向きます。ボーンのローカル座標では+Y軸が腕の方向、+Z軸がこの法線方向となり、+X軸はそれらに直交する右方向（外側方向）になります（右手系を保つため、 $\mathbf{X} = \mathbf{Y} \times \mathbf{Z}$ ）

※注意：肘が完全に伸びきっている場合（肩・肘・手首が一直線上）、上記の外積はゼロに近くなり、法線方向が不安定になります。このときは、上腕骨のロール角は定まらない（自由度が1つ残る）ため、**既知の初期姿勢**（Tポーズ時に手のひらが下向きなら肘の前方向は下方向）を基準にしてツイストを仮定します。または直前フレームの姿勢からロール角を補間することで急激なひねりの跳躍を防ぎます。

- ・**前腕（肘～手首）**：前腕の軸 $\hat{\mathbf{d}}_{forearm}$ は肘から手首へのベクトルです。前腕のロール（橈尺（とうしゃく）回旋）の決定には、手の向きが重要な手掛かりとなります⁴。MediaPipeの出力には手首に加え指先（人差し指・小指・親指の先端）の座標が含まれているため、それらを利用して手の平面を推定します。例えば左手の場合、左手首・左人差し指先・左小指先の3点がなす平面が手のひら平面とみなせます。この平面の法線 \mathbf{n} は

$$\mathbf{n}_{palm} = (\mathbf{p} - \mathbf{p}_{wrist}) \times (\mathbf{p} - \mathbf{p}_{wrist}) - \mathbf{p}_{wrist}$$

のように、人差し指方向と小指方向のベクトルから計算できます（順序は左手と右手で適切に選択します）。 \mathbf{n}_{palm} を正規化したものを前腕の**+Z軸方向**に設定し、これによって「手のひらがどちらを向いているか」を前腕の回転に反映させます。前腕ボーンのローカル+Y軸は肘→手首の方向 $\hat{\mathbf{d}}$ 、そして+X軸はそれらの外積で決まる水平方向（右手系を維持する）となります⁵。こうすることで、手首のひねり（回内・回外運動）が推定されます。手のひら平面が正しく取得できない場合（例えば指先検出が不正確な場合）は、肘～手首と全球座標の上下方向から大まかなひねりを推定したり、直前の姿勢から補間する処置を取ります。 \mathbf{n} は手のひら法線 $\hat{\mathbf{n}}_{palm}$

- ・**手（手首～中手骨基部）**：Humanoidの「Hand」ボーンは手首と指の基準方向をつなぐ短いボーンですが、glTFでは手首そのものがボーンになります。MediaPipeでは指先のランドマークしかないため、手の

ボーン軸は前腕とほぼ同じ方向になります。手ボーンのローカル+Y軸は手首から中指付け根方向と仮定し（モデルのTポーズから算出）、+Z軸は手の甲が向く方向になります。Tポーズでは通常、手のひらは下（手の甲は上）を向いているため、手ボーン+Zは手の甲方向（すなわち前腕ボーン+X軸方向と概ね一致）と定義できます。アニメーション中は、この手の向きは前腕の計算で得た手のひら法線 \mathbf{n}_{palm} に追従させます。つまり手ボーンも前腕同様に、手のひら平面に基づいて回転させます（ただし指先の詳細な傾きは省略）。結果的に、前腕～手先まで一貫した手首のひねり角が表現されます。

- ・ **大腿（腰～膝）**：大腿骨軸 $\hat{\mathbf{d}}_{\text{thigh}}$ は腰（ヒップ左右の中心）から膝への方向です。大腿の曲げ平面は、大腿軸と下腿軸で定義できます。膝の曲がる方向を求めるため、膝→足首のベクトル \mathbf{v}_{shin} を用意し、

$$\mathbf{n}_{\text{thigh}} = \hat{\mathbf{d}}_{\text{thigh}} \times \mathbf{v}_{\text{shin}}$$

と計算した法線を大腿ボーン+Z方向とします。これは膝関節の面の法線であり、膝が曲がる方向の垂直軸を表します。たとえば正面を向いて直立した姿勢では、膝の曲げ平面はおおよそ矢状面（左右方向の面）に近く、 $\mathbf{n}_{\text{thigh}}$ は身体の正面方向を向くでしょう。大腿ボーンのローカル+Y軸は腿の方向（ヒップ中心→膝）、+Z軸は膝の曲げ面法線、+X軸はそれらに直交する方向（外側方向）となります。膝が完全に伸びきった場合には法線が不安定になるため、この場合もデフォルトとして足先の方向（つま先が正面を向くと仮定）を参照して膝のひねりを決めます。

- ・ **下腿（膝～足首）**：下腿骨軸 $\hat{\mathbf{d}}_{\text{shin}}$ は膝から足首への方向です。下腿のロール（脛のひねり）は、足先の向きに密接に関係します。したがって、下腿ボーン+Z方向は足の向きを元に決定します。具体的には、足首・踵・つま先のランドマークから足平面を定義し、その法線を用いて脛の向きを補正します。MediaPipeでは足首（27,28番）、踵（29,30番）、足先（31,32番）にランドマークがあります。左足を例にすると、左踵 \mathbf{p} と左足先 \mathbf{p}_{toe} を結ぶベクトル \mathbf{v} が足の前方方向を表し、これを下腿ボーンの基準平面内に投影して用います。また、足首→踵のベクトルと足首→足先のベクトルから足の平面法線 \mathbf{n}_{foot} を計算し、**下腿ボーン+Z軸**として設定します。こうすることで、足首の回内・回外（足のひねり）が下腿の回転に反映されます。下腿ボーンのローカル+Y軸は膝→足首、+Z軸は $\hat{\mathbf{d}}_{\text{shin}}$ 、+X軸はその外積方向です。例えば直立姿勢で足先が正面を向いていれば、 \mathbf{n}_{foot} はほぼ上向き（足底法線）となり、下腿ボーン+Z軸は上方向に近くなります。歩行や足の傾きに応じて \mathbf{n}_{foot} が変化することで、脛の軸周り回転が推定されます。

- ・ **足（足首～つま先付け根）**：足ボーン（足首から先端）は短く、MediaPipeではつま先先端しかないため、その方向は下腿と重なります。足ボーンの回転は主に足首の屈曲（足先の上下）と足のひねりを担います。足ボーンのローカル+Y軸は足首→つま先方向（ \mathbf{v}_{foot} とほぼ同じ）であり、ローカル+Z軸は足の甲の向きになります。初期姿勢では足の甲は前方かやや上を向いているため、+Zを足の甲法線、+Xを外側方向と定義します。実際の計算では、先に求めた足平面法線 \mathbf{n}_{foot} にできるだけ一致させる）。下腿ボーンとの繋がり、大腿-下腿-足首の曲げ面法線をできるだけ連続にすることで、膝から足首、足首から足先への回転が滑らかに接続されます。 \mathbf{n}_{foot} を参考にして足ボーンの回転を決めます（足ボーン+Zを \mathbf{n}_{foot}

以上のように、末端の各ボーンでは**主軸（+Y）と平面法線あるいは外積から得られる副軸（+Z）**を定義し、それによってボーンのローカル座標系での回転基準を決めています。こうした方法により、関節が曲がる平面上でのボーン前方が一貫して保たれ、捩じり（ロール）成分が推定されます。なお、左右の肢については、外積の順序を適切に反転させ左右対称性を保つ必要があります（例えば右腕と左腕で法線の向きが逆にならないように、常

に身体の前方向に+Z軸が向くよう調整します)。これには各ボーンの初期姿勢における「前方軸」と計算結果を比較し、内積が正になる方向を選ぶ、といった手法で実現できます。

胴体ボーン (Hips, Spine, Chest, Neck, Head) の逐次的な座標系構築

胴体 (体幹) に属するボーンについては、身体全体の姿勢基準フレームを段階的に構築しながら各ボーンの回転を決定します。胴体は複数のボーン (Hips, Spine, Chest, Neck, Head) で構成され、これらは互いに段階的に親子関係を形成しています。胴体の回転計算では、まず骨盤 (Hips) の向きを決め、順次上方へ向かう各ボーン (背骨・胸・首・頭) に対して、その親ボーンの座標系を基準にローカル回転を求める形を取ります。

- **Hips (ヒップ、骨盤)** : Hipsボーンは体幹全体の根元であり、モデル全体の姿勢の基準となります。MediaPipeでは左右の腰 (ヒップ) 点がありますが、モデルのHipsボーン位置は両腰点の中央 $\mathbf{p}_{hipCenter}$ に相当するとします。Hipsボーンの+Y軸は基本的に上方向ですが、骨盤の前傾・後傾や左右の傾きに依りて変化します。ここでは股関節中心→肩中心方向を骨盤のローカルY基準とします。すなわち、 \mathbf{p} (左右肩の中央) を計算し、 $\mathbf{p}_{shoulderCenter} - \mathbf{p}$ をHipsボーンの暫定+Yベクトルとします。このベクトルは上半身へ向かう軸であり、人物が前屈していれば前斜め上を指すなど、胴体の傾きを反映します。

Hipsボーンの+Z軸 (前方軸) は、骨盤の左右方向と上方向から決めます。まず左右の腰点から骨盤の右方向ベクトル \mathbf{v}_{hipRL} を求めます (右ヒップ点 - 左ヒップ点)。これを正規化しローカル+X軸方向とします。次に、グローバルYアップ $(0, 1, 0)$ と \mathbf{v}_{hipRL} の外積をとり、

$$\mathbf{f}_{hip} = (0, 1, 0) \times \hat{\mathbf{v}}$$

と定義します。これにより、水平方向で見たときの骨盤の前方ベクトル \mathbf{f}_{hip} が得られます。人物が水平に回転 (方位) している場合、この \mathbf{f}_{hip} を正規化してHipsのワールド前方ベクトルとします。 \mathbf{f}_{hip} はその人物の向いている方向 (腰の正面) になります ⑥。 \mathbf{f}_{hip}

さらに、骨盤が前後に傾いている場合は、先に求めた $\mathbf{p}_{hipCenter}$ と \mathbf{p} とHips+Y方向 (骨盤上方向) の垂直成分から \mathbf{d} 方向との関係で調整します。具体的には、 \mathbf{f}_{hip} 骨盤平面の法線を計算し、それを用いてHipsボーンの最終的な+Z軸を決めます。Hipsボーンのローカル軸は、+Y軸：骨盤上方向、+Z軸：骨盤前方向、+X軸：右方向 (右手系) となります。

以上でHipsのワールド回転 (基準姿勢に対する姿勢) が定まります。Hipsの回転は身体全体の向き (方位) と傾きを表し、この後のボーン計算の基盤となります。

- **Spine (スパイン、下背)** : Spineボーンは腰から胸にかけての下部脊柱を表します。Spineボーンの親はHipsなので、まずHips座標系内でのSpineの方向を決めます。MediaPipeには背骨中間のランドマークが無いいため、Spineボーンの方角と前方は補間的に決める必要があります。ここでは、先に計算した骨盤座標系を基に、Spineボーンの方角を「股関節中心→肩中心」方向の一部として定義します。例えば、HipsからChestまでを2セグメント (SpineとChest) に分けると仮定し、股関節中心から肩中心へのベクトルを2等分してSpineボーン+Y方向ベクトルとします。すなわち：

$$\hat{\mathbf{d}}_{spine} = \frac{\mathbf{p}}{2} - \mathbf{p}_{hipCenter} \times \frac{\mathbf{p}_{chest} - \mathbf{p}_{hipCenter}}{\|\mathbf{p}_{chest} - \mathbf{p}_{hipCenter}\|}$$

としておき、Spineボーン長に相当するスケーリングを考慮しても方向は同じになります。この $\hat{\mathbf{d}}_{\text{spine}}$ をSpineボーンの軸（ローカル+Y）方向とします。

Spineボーンの前Z軸（前方）は基本的にHipsとChestの中間的性質にします。Hipsボーンの前 \mathbf{f}_{hip} と、後述するChestボーン前 $\mathbf{f}_{\text{chest}}$ との間で補間するのが理想ですが、Chest計算前の段階では $\mathbf{f}_{\text{chest}}$ は未知です。そこで近似として、**Hipsの前方ベクトル**をそのままSpineボーンの前方とします。すなわち、Spineボーンの前Z方向 $\hat{\mathbf{f}}$ を $\hat{\mathbf{f}}_{\text{hip}}$ と同一視します（人物が腰と肩で大きくねじれている場合は、ねじれの半分をSpineに分配する方法もありますが、ここでは簡略化します）。SpineボーンのローカルZ軸を $\hat{\mathbf{f}}$ 、+X軸をそれらの外積方向（右向き）とすれば、Spineのワールド回転が定まります。**胴体のねじれ**がある場合（腰と肩の向きが異なる）、その差分は後のChestボーンに反映させます。

- **Chest（チェスト、上背）**：Chestボーンは上部脊柱で、肩付近の姿勢を担います。MediaPipeでは左右の肩座標が得られるため、Chestボーンはそれらの中点 $\mathbf{p}_{\text{shoulderCenter}}$ に位置すると考えられます。Chestボーンの前Y軸はSpineボーンと同様に胴体上方向ですが、SpineとChestの2セグメントモデルでは、Chest+Yもまた $\mathbf{p}_{\text{shoulderCenter}}$ 方向と概ね一致します（SpineとChestで同じ方向になるため、実質的に背骨全体を一直線とみなす近似です）。厳密には、SpineとChestにそれぞれ腰から肩へのベクトルの何割かを割り当てる形で**分割**すると良いでしょう。例えばSpineが腰から胸の2/3、Chestが残り1/3の長さを持つとした場合、それに比例してSpineとChestの傾きを分けます。ただし簡潔のため、ここではChest+Yも $\hat{\mathbf{d}}$ とほぼ同じとします。

Chestボーンの前Z軸（前方）は肩の向きを使って決定します。左右肩の位置から**肩ラインの向き** $\mathbf{v}_{\text{shoulderRL}}$ を計算できます（右肩 - 左肩）。これを正規化して**肩の左右軸**とします。次に、グローバルY軸（上）との外積で**肩ベースの前方**を求めます：

$$\mathbf{f}_{\text{chest_horiz}} = (0, 1, 0) \times \hat{\mathbf{v}}.$$

これは水平面上で見た肩の正面方向を表します。人物が上半身だけ左右にねじっている場合（肩の向きが骨盤と異なる場合）、 $\mathbf{f}_{\text{chest_horiz}}$ は骨盤前方 \mathbf{f}_{f} とは異なる方向になるはずですが、その差が胴体のねじれ角です。

さらにChestの前後の傾き（前かがみ／反り）を反映するため、先ほどの肩中央→首（または肩→腰）方向との関係で**胸部の傾き**を考慮します。肩から首のベクトル $\mathbf{v}_{\text{neckBase}}$ （後述のNeck計算で使用）や肩と股関節の高さ差などから、胸が前方に傾いている場合は \mathbf{f}_{f} を基準にします。 \mathbf{f} をやや下向きに補正します。しかし細かな傾斜はNeck/Headで微調整されるので、Chestの前方はまず水平成分 $\mathbf{f}_{\text{chest_horiz}}$

以上から、Chestボーンの前Z軸は $\hat{\mathbf{f}}_{\text{chest}} = \mathbf{f}_{\text{f}}$ と一致）となります。ChestボーンはHipsからNeck基部までの姿勢を担い、ここで胴体の水平方向のねじれ（腰と肩の方位差）が反映されます。例えば腰が正面を向いていても肩が左を向いていれば、HipsボーンとChestボーンの前ベクトルが異なり、その差分ねじれがChestの回転に表現されます。 \mathbf{f} （正規化）、ローカル+Y軸は胸の上向き（Spineとほぼ同方向）、ローカル+X軸は右肩方向（ $\mathbf{v}_{\text{shoulderRL}}$

- **Neck（ネック、首）**：Neckボーンは首の付け根から頭蓋底までの短い骨です。MediaPipeには首のランドマークはありませんが、左右の肩中央と鼻の位置や左右耳の位置から首の姿勢を推定できます。まず、**首の位置**を肩中央と頭（耳や鼻）の中間付近と仮定し、例えば肩中央点 $\mathbf{p}_{\text{shoulderCenter}}$ と

鼻先 \mathbf{p} としましょう。 \mathbf{p} の線を引き、その $\sim 80\%$ 程度の位置を首の基部（Neckの子ボーンであるHeadの位置）とします。この推定点を $\mathbf{p}_{\text{neckBase}}$

Neckボーンの+Y軸はChest→首基部方向、つまり $\mathbf{p}_{neckBase}$ - \mathbf{p} ベクトルの正規化 $\hat{\mathbf{d}}_{neck}$ とします。Neckボーンの+Z軸（前方）は、Chestボーンの+Zと頭部の向きの間です。頭部の向きは後述のHeadボーンで決めますが、首ではその**可動範囲の一部**を負担するようにします。例えば、胸と頭の向きが異なる場合、首ボーンは全体のねじれ・傾きの30~50%程度を担い、残りを頭ボーンで処理すると自然です。そのため、Neckの前方ベクトル \mathbf{f} はChest前方 \mathbf{f}_{chest} と頭目標前方 \mathbf{f} （後述）の線形補間などで決めます（オフライン処理であれば、Quaternionの球面線形補間(SLERP)で首と頭の回旋を分割する手法も取れます）。ここでは概念説明として、Neck+ZはChest+Zにほぼ揃え、頭の向きは主にHeadボーンで追従させるものとします。

まとめると、Neckボーンのローカル+Y軸は首の上方向（胸から頭へ方向）、+Z軸は胴体の前方（若干頭寄りに補正可能）、+X軸は右向き（肩と頭の横方向の中間）となります。Neckは胸と頭をつなぐ緩衝として機能し、大きな上下動作（うなずき等）は頭ボーンで表現しつつ、細かな前傾やねじれを分担します。

- Head (ヘッド、頭)** : Headボーンは首の上端から頭頂または顔の基準方向へのボーンです。MediaPipeから得られる頭部の情報として、鼻先(ランドマーク0)、両目(1~6番)や耳(7,8番)が利用できます。ここでは鼻先と左右耳の位置から頭部の向きを推定します。まず、左右耳を結ぶベクトル \mathbf{v}_{earRL} を計算し、頭部の左右軸(ローカル+X方向に相当)とします。次に、耳の中点と鼻先を結ぶベクトル \mathbf{v} を計算します。これは顔が向く大まかな方向ベクトルで、頭部の**前方**を表します。頭ボーン**の+Z軸**はこの \mathbf{v} を正規化したもの $\hat{\mathbf{v}}$ とし、**+Y軸**は頭頂方向を決める必要があります。頭頂(上向き)は顔の前方と左右軸に直交する方向なので、 $\hat{\mathbf{v}} \times \mathbf{v}_{earRL}$ (頭の左右方向)と $\hat{\mathbf{v}}$ (頭の前向き)の外積を取り：

$$\mathbf{u}_{head} = \hat{\mathbf{v}} \otimes \hat{\mathbf{f}}_{head}$$

これを正規化した $\hat{\mathbf{u}}_{\text{head}}$ を頭ボーンの+Y軸（頭の上方向）とします⁷。最後に、ローカル+X軸は $\hat{\mathbf{f}}$ で計算できます（右手系の条件を満たすよう順序に注意）。こうして得られたHeadボーンの回転は、顔が向く方向と頭の傾きを反映しています。例えば鼻先がやや上を向いていれば頭ボーンは上向きに回転し、左右の耳の位置関係から首傾げ（左右の傾き）も反映されます。

頭部の回転はNeckボーンの親子関係で最終調整されます。前述のように、胸～頭の向き差をNeckとHeadで分担する場合、Headボーンの目標前方 $\hat{\mathbf{f}}_{\text{head}}$ は実際の鼻先ベクトルそのものですが、Neckボーンが既に胸から一部追従している場合には、Headボーンでは残差のみ回転する形になります。オフライン処理であれば、Headボーンの回転を直接計算した後、Neckボーンの回転と合成して頭部全体として正しい方向になるように補正することも可能です。

以上、胴体ボーンでは**身体軸に沿ったY方向と体の正面方向を表すZ方向**を各段階で定め、順次上半身へ姿勢を伝搬させています。Hipsで決めた水平面上の方位はChestまでに徐々に反映され、Neck/Headで最終的な頭の向きになります。胴体の各ボーンのローカル座標系は、いずれも+Y軸が上方向（各セグメントの軸）、+Z軸が前方向（正面）で揃えられており、モデルの初期Tポーズではこれらがワールド座標系の対応軸とほぼ一致するようにデザインされています。

ターゲットワールド回転の構築とローカルQuaternionへの変換

上述の手順で得られた各ボーンの目標となるワールド座標系での軸（+Y方向と+Z方向）に基づき、各ボーンが最終的に持つべきワールド空間回転（Quaternion）を構築します。具体的には、ボーンのワールド座標系において

- +Y軸を先に求めたボーン軸方向 $\hat{\mathbf{d}}$ に一致させ、
- +Z軸を各ボーンごとに定義した前方ベクトル $\hat{\mathbf{f}}$ に可能な限り一致させ（Y軸と直交するように調整）、

残る+X軸は右手系を形成するよう外積で算出します⁵⁷。この3本の直交軸から回転行列を組み立て、それをQuaternionに変換することでボーンのワールド回転を得ます。例えば、各ボーンについて以下のように行列の列ベクトルを配置します（UnityやOpenGLの慣習ではZを前方、Yを上、Xを右とするためそれに準じます）⁷：

- 第1列：ボーンの右方向ユニットベクトル（ローカル+X軸 = $\hat{\mathbf{d}} \times \hat{\mathbf{f}}$ ）
- 第2列：ボーンの上方向ユニットベクトル（ローカル+Y軸 = $\hat{\mathbf{d}}$ ）
- 第3列：ボーンの前方向ユニットベクトル（ローカル+Z軸 = 上記2列の外積で再計算された正規直交ベクトル）

このようにして得られた回転行列は、ボーンがワールド空間で目標とする姿勢を表します。行列をQuaternionに変換する手順は標準的なもの（例えば回転行列のトレースから計算⁵）を用いますが、ライブラリ関数を使うこともできます。

ここで重要なのは、各ボーンのワールド回転はモデルのTポーズ（バインドポーズ）に対する変化だという点です。したがって、単にワールド回転を求めただけではアニメーション適用できない場合があります。次項で、バインドポーズからの差分としてローカル回転を得る方法を説明します。

バインドポーズとの差分計算とローカル回転Quaternionの算出

モデルの各ボーンには、glTFのバインドポーズにおける初期回転（Bindposeのローカル回転）が存在します。この初期姿勢からMediaPipe由来の目標姿勢への差分を計算し、各ボーンのローカル回転値として出力するのが最終ステップです。基本的なアプローチは「現在のボーンのワールド回転」を「初期ボーンのワールド回転」で逆変換することです。

1. **ボーンの初期ワールド回転**：まず、モデルのTポーズにおける各ボーンのワールド空間での回転 $Q_{\text{bind}}^{\text{world}}$ を求めます。これは、glTFに定義されたボーンの初期ローカル回転（および親ボーンからの階層的な変換）から計算できます。多くのモデルではTポーズ時にボーンのローカル回転は零（アイデンティティ）になっており、初期ワールド回転は親ボーンの変換に一致しますが、モデル次第では初期ローカル回転が入っている場合もあります。いずれにせよ、ボーンの初期姿勢での方向ベクトル（例えば上腕なら肩→肘の方向）をもとに、先ほどと同様な軸セット（+Y方向＝骨方向、+Z方向＝骨の前方基準）を構築すれば $Q_{\text{bind}}^{\text{world}}$ が得られます。

2. **ボーンの目標ワールド回転**：前段までで算出した、MediaPipeのデータから導出された各ボーンの**目標ワールド回転**を $Q_{\text{target}}^{\text{world}}$ とします。このQuaternionは、ボーンがワールド空間で取りうる最終的な向きを表します。

3. **差分（補正）回転の算出**：ボーンを初期姿勢から目標姿勢へ回すクォータニオン Q_{Δ} は、

$$Q_{\Delta} = Q_{\text{target}}^{\text{world}} \cdot \left(Q_{\text{bind}}^{\text{world}}\right)^{-1}$$

として求められます³。すなわち、初期ワールド回転の逆を右から掛けることで、初期の向きをキャンセルし目標への回転だけを抽出します。この Q_{Δ} こそが、ボーンの**ワールド空間での補正回転**（初期姿勢から見た回転差）です。

1. **ローカル回転への変換**：しかし、最終的に必要なのは**各ボーンのローカル座標系での回転**です。親ボーンも回転しているため、子ボーンのローカル回転はワールド差分をそのまま使えません。一般に、子ボーンの新しいローカル回転 $Q_{\text{new}}^{\text{local}}$ は、**親ボーンの現在のワールド回転** $Q_{\text{parent}}^{\text{world}}$ を使って、

$$Q_{\text{new}}^{\text{local}} = \left(Q_{\text{parent}}^{\text{world}}\right)^{-1} \cdot Q_{\Delta}$$

と求められます。ここで $Q_{\text{parent}}^{\text{world}}$ は親ボーンについて一つ前のステップで計算済みの $Q_{\text{target}}^{\text{world}}$ を用います。つまりルート（Hips）から順に、各ボーンのワールド回転を計算しつつ、同時にローカル回転へと変換していきます⁸。例えば、上腕ボーンのローカル回転はChestボーンのワールド回転の逆でターゲット上腕ワールド回転を挟むことで算出します。結果として得られる $Q_{\text{new}}^{\text{local}}$ が、モデルに適用すべきそのボーンの新しいローカル回転値です。

もう一つの見方として、初期ローカル回転 $Q_{\text{bind}}^{\text{local}}$ が既知であれば、直接的に**ローカル差分**を求めることもできます。ボーンの目標ローカル回転は

$$Q_{\text{new}}^{\text{local}} = Q_{\text{bind}}^{\text{local}} \cdot Q_{\Delta}$$

と表せます。ただし注意点は、 Q_{Δ} をボーンのローカル空間における差分とみなすためには、 Q_{Δ} をいったん親ボーンの座標系に取り込む必要があることです。現実装では前述のワールドからローカルへの変換式の方が間違いが少ないでしょう。

以上のステップにより、各ボーンの**ローカル回転Quaternion**が揃います。このQuaternionはglTF Humanoidの各ボーンに適用可能な回転値で、MediaPipeの33点ランドマークで表されたポーズを正確に再現するものです。特に姿勢がTポーズから大きく変化している場合でも、バインドポーズとの差分をとることでモデル固有の初期姿勢の影響を打ち消して正しい変換が得られます。

ベクトル・行列・Quaternion変換の総合ステップと誤差対策

最後に、全体のアルゴリズム手順をまとめ、計算上の留意点（誤差要因とその対処法）について述べます。

全体の処理フロー ③：

1. **入力データ変換**：MediaPipeの33関節ワールド座標をモデルの座標系に変換する ① ②（原点と軸の調整）。
2. **ボーン方向計算**：各ボーンについて親関節→子関節のベクトルを計算し正規化（ボーン軸=ローカル+Y）。胴体では親子間を補間して定義。
3. **参照ベクトル計算**：各ボーンごとに平面法線や外積などから第二のベクトルを計算（ボーン前方=ローカル+Z）。必要に応じて初期基準に合わせ符号を揃える。
4. **直交座標軸構築**：ボーン軸(+Y)と前方(+Z)が直交しない場合、前方ベクトルをボーン軸に直交する平面に射影して調整する。次に右方向(+X)を $\mathbf{X} = \mathbf{Y} \times \mathbf{Z}$ で算出する ⑤。こうしてボーン固有の基底（右手系の回転行列）を得る ⑦。
5. **Quaternion化**：得られた基底からボーンのターゲットワールド回転 $Q_{\text{target}}^{\text{world}}$ をQuaternionに変換。
6. **差分計算**：各ボーンについて、 $Q_{\text{new}}^{\text{local}} = (Q_{\text{parent}}^{\text{world}})^{-1} \cdot Q_{\text{target}}^{\text{world}}$ を計算し、新ローカル回転を得る（ルートのHipsは親が無いので直接 $Q_{\text{target}}^{\text{world}}$ がローカル回転になる）。同時に $Q_{\text{parent}}^{\text{world}}$ は親ボーンの $Q_{\text{target}}^{\text{world}}$ （ターゲットワールド回転）を使用。
7. **出力**：17ボーンそれぞれのローカル回転QuaternionをHumanoidのボーン順に出力。

この際に留意すべき誤差要因と対処法：

- ・ **ツイストの非一意性**：前述の通り、関節がまっすぐ伸びた状態では骨の軸周りの回転が不定になります（例えば肘や膝が伸展した状態、あるいは背骨を垂直に伸ばした姿勢では捩じりが判別困難です）。この場合、急激なロールの揺れを防ぐために**デフォルトのロール角**を決めておきます。Tポーズを基準に「この姿勢ではひねりゼロ」とみなす方向を各ボーンについて定義し、計算された前方ベクトルがそれと鋭角になるよう符号を選択します。例えば上腕なら、Tポーズで肘の前方は真下と仮定すれば、肘が伸びたときは上腕ボーンのローカル+Zが地面方向に近くなるようにロールを初期化します。
- ・ **平面法線計算の不安定性**：関節点のノイズや一直線配置により外積で得られる法線ベクトルが不安定になる場合、**閾値処理**や**フィルタ**を用います。具体的には、外積ノルムが小さい（例えばある角度閾値未満の鋭角または鈍角で3点がほぼ一直線）場合には、直前フレームの法線を流用したり、法線をグローバル軸（例えば腕ならグローバル前方など）にスナップさせることで急激な向き反転を避けます。オフラインであればSavitzky-GolayフィルタやKalmanフィルタでランドマークのばらつきを平滑化することも有効です ⑥。
- ・ **符号一貫性の維持**：外積によって得られるベクトルには符号のあいまいさ（方向が逆でも同じ平面を表す）があるため、各ボーンの基準軸について**一貫した方向**を選ぶ必要があります。これは、初期Tポーズでの各ボーンのローカル軸（+Zが向いていた方向など）と計算結果を比較し、内積が正になるように外積ベクトルを場合によっては反転する処理で対処します。または、親子関係のボーン同士でねじれ方向が連続するよう、親ボーンの軸に対する子ボーンの参照ベクトルの投影符号を調整します。
- ・ **骨長の不整合と縮尺**：MediaPipeの3D座標は実測単位（メートル）で出力されますが ⑨、モデルの骨長とは必ずしも一致しません。そこで、事前にモデルの各ボーンの長さを計算し、MediaPipeの各フレームで得られるジョイント間距離をその長さに**正規化**または**スケーリング**します ⑩。これにより、たとえMediaPipeの検出位置に誤差があっても、出力するボーン姿勢はモデルの寸法に合致し、**スケーリング誤**

差による回転ズレ（例えば肘が届かない/オーバーシュートする問題）を防げます。また、不要なスケーリング要素はQuaternionには含めず回転のみに集中できるようになります。

- ・**階層的な誤差伝播**：親ボーンの僅かな回転誤差が子ボーン先端で大きく見えることがあります。これを抑えるため、重要な基準（例えば足底が地面につく、視線方向を維持する等）がある場合は、そのボーンに対し**逆運動学（IK）的な補正**を加えることも考えられます。ただし本アルゴリズムはオフライン前提なので、必要ならポスト処理でIK補正し、再度各ボーンのQuaternionを調整すると良いでしょう。

以上、OpenGL座標系におけるMediaPipe33点からHumanoid17ボーンへの回転変換アルゴリズムを詳細に述べました。各ボーンについて、ワールド空間での目標姿勢（+Y軸方向と+Z軸方向）を求め、それを基にQuaternionを算出し、バインドポーズとの差分としてローカル回転に落とし込む一連の工程を解説しました³。適切な座標変換と基準軸の設定により、MediaPipeの推定ポーズをモデルの骨格に正確に適用でき、ひねりや関節面の不整合も最小化できます。各ステップで発生しうる曖昧さ（ツイストの自由度や法線計算の不安定さ）についても、初期姿勢の基準や平滑化によって対処可能であることに触れました。以上の方法を実装することで、リアルタイムでなくともオフライン処理で高精度なHumanoidボーンのQuaternion姿勢を得ることができるでしょう。

参考文献：MediaPipe提供の座標系仕様¹、glTFの座標系定義²、Jose M. Restrepo氏によるモーションキャプチャパイプライン解説³、およびQuaternionと行列の関係に関する知見⁵⁷などを総合して本アルゴリズムを構築しました。

¹ 3D Pose Detection with MediaPipe BlazePose GHUM and TensorFlow.js — The TensorFlow Blog

<https://blog.tensorflow.org/2021/08/3d-pose-detection-with-mediapipe-blazepose-ghum-tfjs.html>

² glTF™ 2.0 Specification - Khronos Registry

<https://registry.khronos.org/glTF/specs/2.0/glTF-2.0.html>

³ ⁶ ⁸ ¹⁰ Introducing BAPH: Building an End-to-End Markerless Mocap → BVH Pipeline

<https://www.linkedin.com/pulse/introducing-baph-building-end-to-end-markerless-mocap-restrepo-d8w0f>

⁴ ⁵ ⁷ mathematics - Quaternion representing rotation from shoulder to hand - Game Development Stack Exchange

<https://gamedev.stackexchange.com/questions/165143/quaternion-representing-rotation-from-shoulder-to-hand>

⁹ python - Transform value 3D coordinate pose landmarks (Mediapipe) to real world value in meters? - Stack Overflow

<https://stackoverflow.com/questions/72185599/transform-value-3d-coordinate-pose-landmarks-mediapipe-to-real-world-value-in>