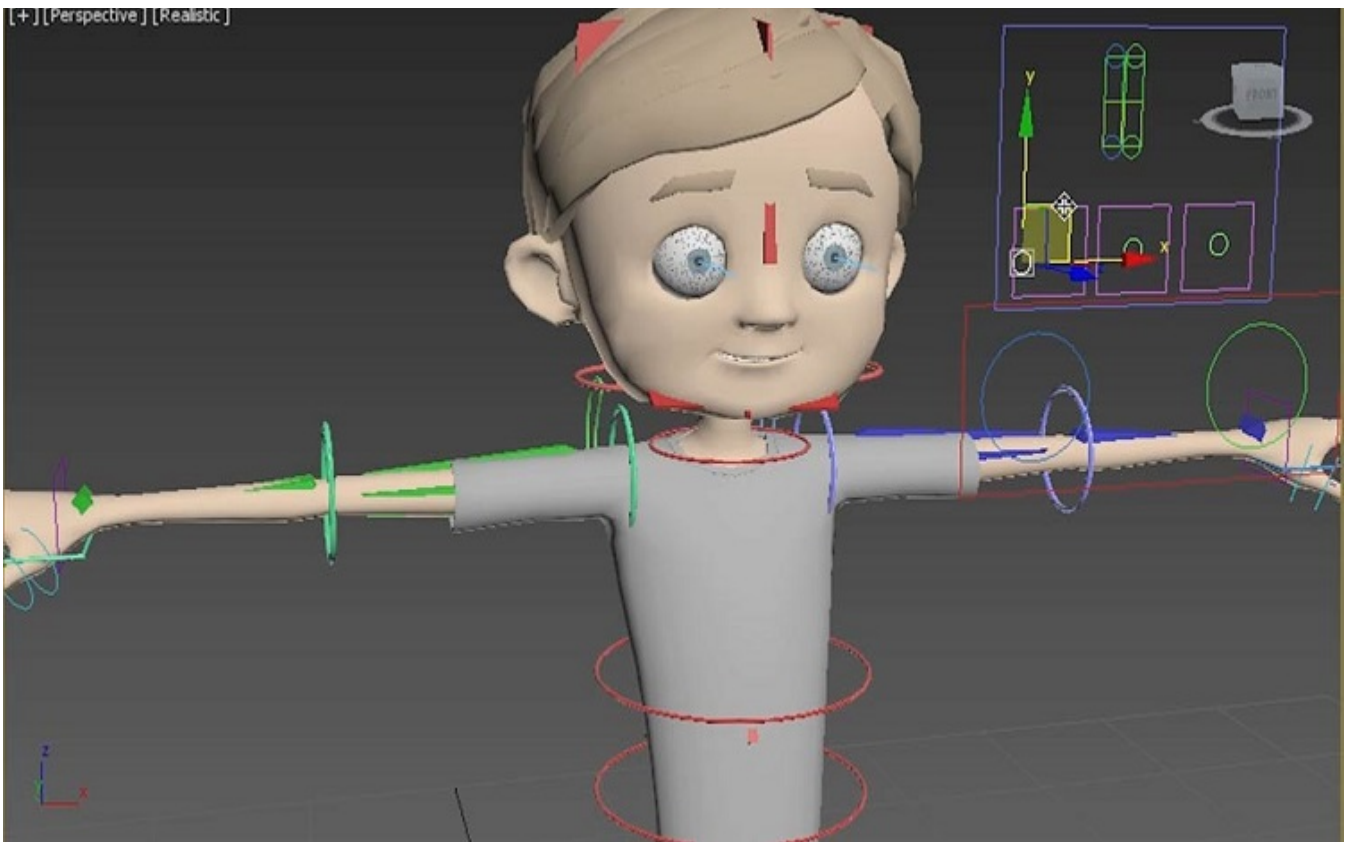


キャラクター・アニメーション: スケルトンと逆運動学

この記事は、インテル® デベロッパー・ゾーンに公開されている「[Character Animation: Skeletons and Inverse Kinematics](#)」の日本語参考訳です。

英語のオリジナルの記事は、VentureBeat で「[Character Animation: Skeletons and Inverse Kinematics](#)」として公開されています。インテルのゲーム開発のニュースと関連トピックは、[VentureBeat \(英語\)](#) を参照してください。

Steve Pitzel



はじめに

ゴシック様式のお城が完成したら、ヒーローを連れてきて門を守らせましょう。ゲームエンジンは、スケルトン、逆運動学 (IK) と呼ばれるアニメーション・システム、制御用の複雑な階層セットアップをサポートしています。すべての機能を活用しようとするのは良いことです。しかし、以前にスケルトンを使用したり、関節があるモデルに IK を利用したことがなければ、驚くことになるでしょう。それらのいくつかは上手く実装できるでしょうが、いくつかは (少なくとも初めは) 上手く実装できないかもしれません。

この記事では、逆運動学および順運動学 (アニメーションするキャラクターを動かすためのトップダウンの回転システム) の両方を含むスケルトンの利用について、いくつかの原理を説明します。その途中で、いくつかの

質問に答えています。この記事を読み終えた後に、おそらく読者の皆さんは現在の問題を乗り越えられるようになっていくでしょう。

順運動学と逆運動学

つい最近まで、ゲームのキャラクターは岩のように硬いセグメントのかたまりであり、滑らかに動かすことは困難でした。当時のアニメーターが望んでいたのは、基本的な階層を処理でき、一度にすべてのセグメントの位置を決定してキーフレームにできるプラットフォームでした。現実的に肘を曲げたり筋肉を収縮させることは、実際に再現できない以上、思い出さないようにしました。現在利用可能な高い処理能力なしでリアルタイムに表面の変形を制御するため、多くのゲームはロボットや装甲を特色としていました。

最新のプロセッサの高い処理能力で逆運動学を使用した CG スケルトンのアニメーションが可能に

しばらくの間、ゲーム開発者たちは無言で苦しんでいました。その後、PC が高速かつスマートになり、ゲームエンジンも高速かつスマートになりました。処理性能の向上により、アニメーターは、コンピューター生成 (CG) スケルトンを使用して、セグメント化されたキャラクターを 1 つにまとめるだけでなく、キャラクターの皮膚を実際に変形できるようになりました。IK は多くのアニメーターが選択する制御システムになりました。

階層について

逆運動学を理解するには、基本的な階層構造と順運動学を理解することが重要です。ターゲットからターゲットへのモーフィングを除いて、アニメーターは、ほとんどの場合、何らかの階層を使用してキャラクターのアニメーションを行っています。階層は、親-子-兄弟関係です。オブジェクトを親または別の子として割り当てるプロセスは、「親子関係の作成」と呼ばれます。「グループ化」と呼ばれることもあります。人間の脚の場合、脚の上部の親が腰に、脚の下部のが脚の上部の子に、足が脚の下部の子になります。右脚上部は左脚上部と兄弟関係を共有し、どちらも腰の子になります (図 1 を参照)。

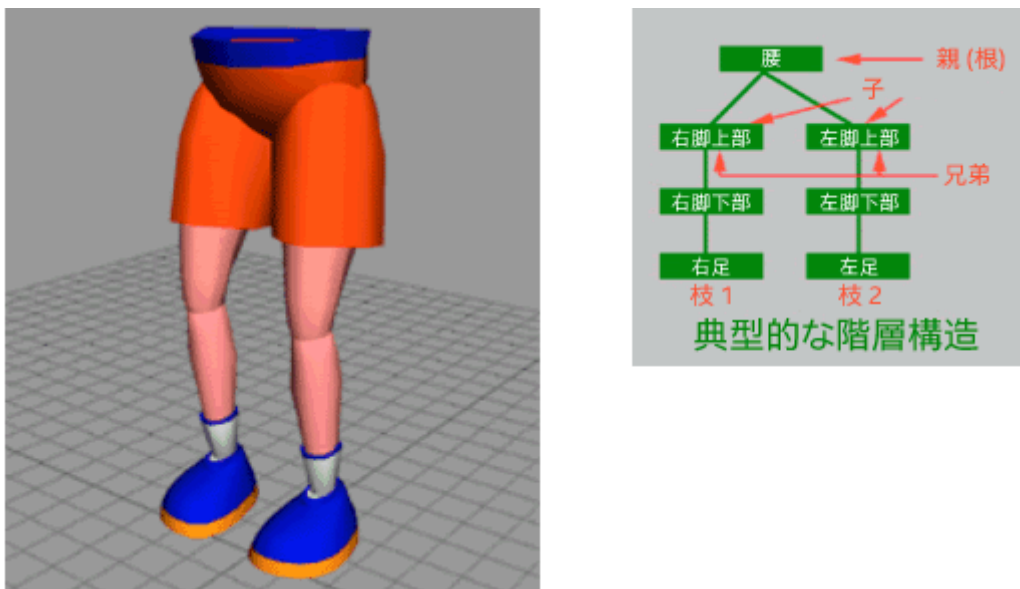


図 1. 基本的な下半身のセットアップと順運動学の階層構造

別の形式の階層構造は、階層の親 (腰や骨盤) が「根 (ルート)」となる「逆ツリー」モデルを使用します。階層の一部が「幹」になることはなく、子、およびその子は「枝」になります (図 1 を参照)。

順運動学の問題点

基本的な階層構造の下でキャラクターの手を動かす場合、最初に上腕を回転し、次に前腕を回転して、最後に手を回転します。この「トップダウン」の回転システムは順運動学と呼ばれ、基本的なアニメーションではうまく動作します。

アニメーションは、順運動学と基本的な階層構造では単純に見えます。キャラクターが歩くような動きをする必要があるまで、床の上で足を動かすことはありません。後ろを見るために身体を回している間も、その場にとどまっています。この場合、2つの選択肢がありますが、どちらも美的に優れているとは言えません。

- 階層全体を回転します (脚も回転する必要があります)。
- ほかの部分はそのままで階層の上部 (腰) のみ回転します (見るに堪えません)。

この時点で、順運動学 (およびキャラクター) はがらくたになってしまいます。階層の下部のオブジェクトにとって最も困難なことは、階層の上のオブジェクトが動いているときに同じ場所にとどまることです。このため、歩いているキャラクターの足を適切に処理することはほぼ不可能です。動いていないと錯覚させるには、一定の再調整が必要です (図 2 を参照)。

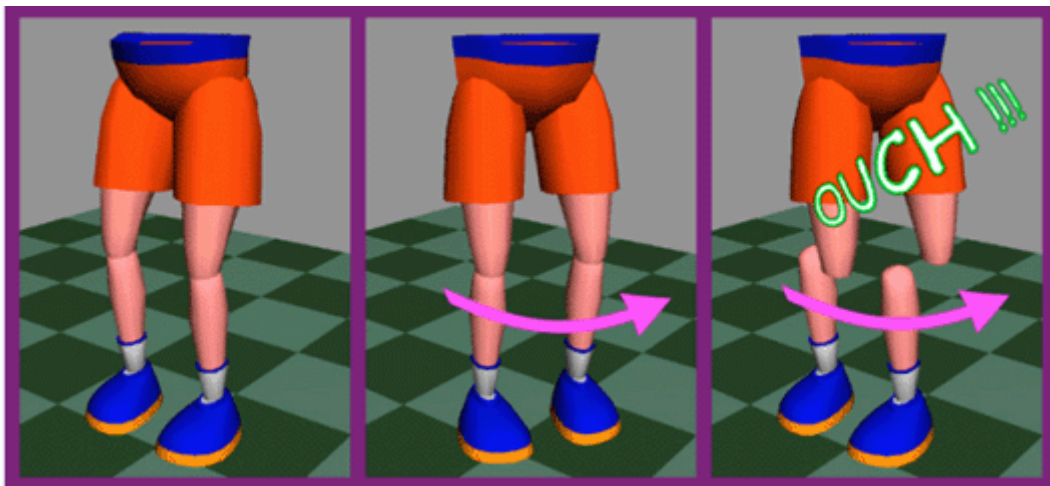


図 2. 基本的な階層構造と順運動学では階層全体を回転するときにも足も回転する必要があり、足を回転しないと見るに堪えない結果になる

逆運動学を使用した CG スケルトンの登場

幸いなことに、現在のゲーム・アニメーターは、かつてアニメーション映画でのみ利用されていたアニメーション手法の多くを利用することができます。アニメーション映画の世界では、足首、膝、腿、腰のわずかな動きを補うために、恐竜のつま先を絶えず再調整して、(ピクセル単位ではなくフィート単位で) スクリーン上を滑らせています。キャラクターのアニメーションをより自然に行うには、順運動学以外の何かが必要であることが明らかになりました。別の映画の問題 (キャラクターが動くときに関節を折り曲げたり収縮させる描写に真実味を持たせる) は、連続するメッシュから作成されたモデルを使用して解決できそうです。では、固まりを折り曲げたり歩かせるにはどうすれば良いのでしょうか。初期の映画の特殊効果の代わりに、Stop Motion* ソフトウェア開発者はアーマチュア・システム (スケルトン) を CG モデルに組み込み、スケルトンを制御する逆運動学と呼ばれるシステムを開発しました (図 3 を参照)。

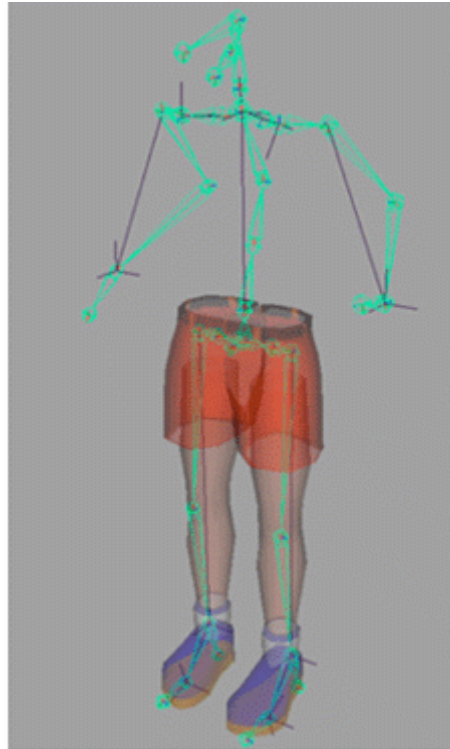


図 3. CG スケルトン

概念的に、CG スケルトンは非常に分かりやすいものです。人間の物理的な骨を模倣しており、すべての骨を保持する関節に腱のある厳密な構造です。逆ツリー階層であり、実際に親はスケルトンの「ルート」と呼ばれます。ルートは通常、キャラクターの自然な重心に位置します。二足動物では、背骨の基点になります。このシステムを基本的な (非骨格) 階層よりも優れたものにするのは腱ですが、コンピューター 3D プログラムでは見えません。非骨格オブジェクトから構築された階層のように、スケルトンは順運動学によりトップダウンで操作されます。しかし、スケルトンの最も大きな利点は、階層の上部から下部へ制御を移動するその能力にあります。ボトムアップ (順運動学の反対) で動作または「解決」するため、このプロセスは「逆運動学」または IK と呼ばれます。このプロセスは、トップダウンの回転に取り組むよりも直感的であり、大幅に時間を節約できます。

注: 一部のアニメーション・パッケージでは、立方体、球体および Null オブジェクトのような「非骨格」オブジェクトに逆運動学のアプリケーションを作成できますが、この方法による IK チェーンの作成や操作は通常よりはるかに複雑です。

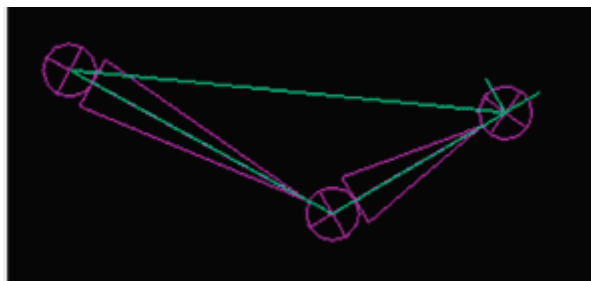


図 4. IK ハンドルを利用した単純な骨のセットアップ。骨はルートの終点の近くでは広く、エフェクターの近くでは狭く描かれます。肘を曲げることにより、肩および手首との間で形成された緑の三角形は、肘の回転平面を示しています。この平面は、肘を外に回せるように調整されます。

骨格チェーンの一番下のリンクは、ゴールまたはエンド・エフェクターと呼ばれます (図 4 を参照)。(これらの名前は、異なるプログラムでは異なるものを意味することがある点に注意してください。)セグメント化されたモデルでは、キャラクターの「肉」、実際の手足、胴、頭などは、スケルトンの最も近くの骨 (または関節) との親子関係が作成されます。単一メッシュの多角形またはシームレスな NURBS モデルでは、スケルトンに対してモデルの「スキン」または「エンベロープ」を設定します。

注: キャラクターの手足、頭、皮膚、衣服は「ジオメトリー」と呼ばれます。スケルトンは通常レンダリングできませんが、ジオメトリーはレンダリングできます。

スケルトンの場合、ジオメトリーをアニメーションする代わりに、スケルトンそのものをアニメーションして、ジオメトリーはそれに従います。骨と骨の間には見えない腱があるため、階層の一部のみが回転しても関節は離れません。また、骨格チェーンに IK を適用できます。

逆運動学の利点

これらはすべて、アニメーターにとって何を意味するのでしょうか。これは、チェーンの下部 (例えば、手) を引っ張り、手の上のすべての骨を自動的に回転して適切な位置に配置できることを意味します。これは、例えば、木からりんごを取るように、キャラクターが何かに手を伸ばさなければならない場合に非常に役立ちます。手を適切な位置に配置するには肩、上腕、肘、手首をどのくらい回転すれば良いのか考えて計算するのではなく、アニメーターがりんごの上に手を置くだけで、腕の残りが続きます。

スケルトンを使用するもう 1 つの利点は、骨と関節により表面を変形する自然な制御構造が提供される点です。変形の際に、ジオメトリーの個々の制御ポイントは骨および関節と関連して移動します。これはジオメトリーの部分全体を一度に移動するよりもはるかに効率的です。

スケルトンに対してモデルの「スキン」または「エンベロープ」を設定すると、逆運動学または順運動学のいずれかを使用してアニメーションを行うことができます。スケルトンのいくつかの部分では、順運動学を使用する利点がまだあります。一部のパッケージでは、同じ骨格チェーンで順運動学と逆運動学の両方を使用して、瞬時に切り替えることもできます。ほかのパッケージでは、膨大な労力をかけた場合にのみ切り替えることができます。この切り替えを行うには通常、コンストレイント (キーフレームのオンオフを切り替えることができる磁力と考えます)、式 (数式)、または両方の組み合わせを使用する必要があります。

骨の基本

まずソフトウェア・パッケージについて見た後、自分の骨と関節について見て、最後にそれらがどのように制御されているか見ていきましょう。

ソフトウェア・パッケージ

スケルトンは、すべてのアプリケーションで同じ方法で表現あるいは制御される訳ではありません。IZWare Mirai*、Alias Wavefront Maya* および PowerAnimator* などの一部のパッケージでは、手足が分岐した、完全に接続されたスケルトンを作成することができます。この記事の公開時点では、Softimage* 3D と XSI では同様のスケルトンは作成できませんが、親子関係を作成して腕チェーン、脚チェーンおよび首チェーンをとともに組み合わせる手段や「コンストレイント」と呼ばれる個別の制御システムが提供されています。

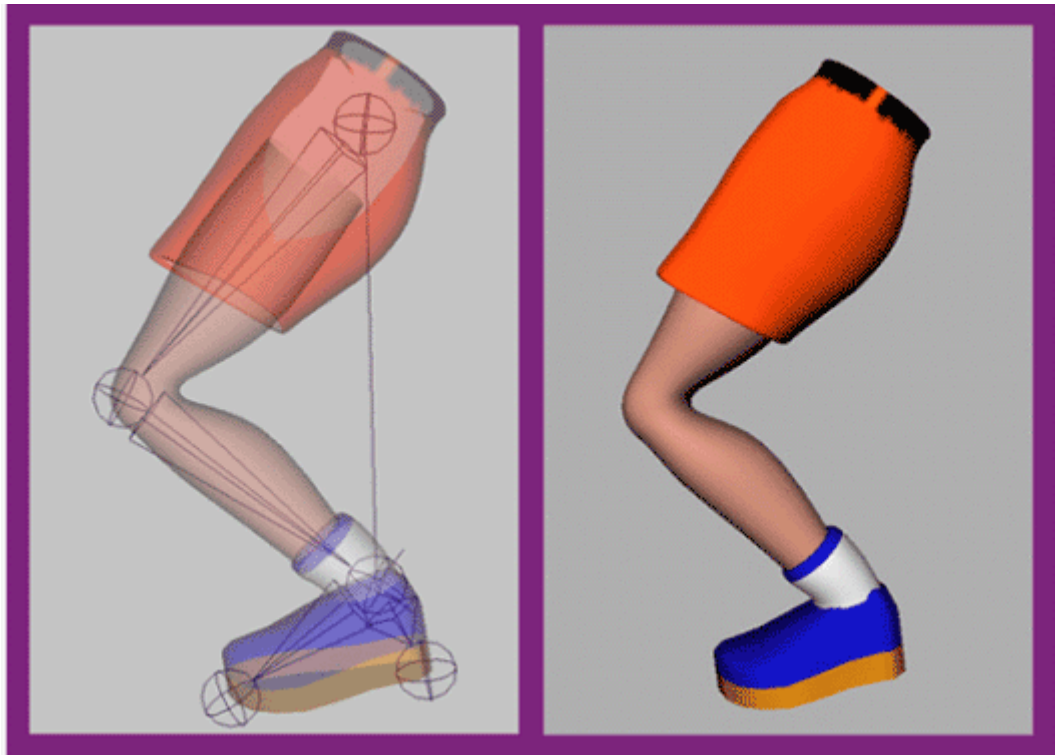


図 5. ジオメトリーに「スキン」または「エンベロープ」を設定すると、表面の制御ポイントは最も近い骨により制御されます。骨が表面の制御を共有すると、非常に滑らかな抱え込みや折り曲げが可能です。表面の変形が希望する結果でない場合でも、スケルトンはセグメント化されたモデルを保持し、関節ですべての部分に自然なピボットポイントを提供できることは大きな利点です。

一部のパッケージでは、逆運動学システムは使用できますが、骨は使用できません。代わりに、基本オブジェクトから独自のスケルトンを作成するか、各キャラクターの個々のパーツに骨のプロパティを割り当てます。Softimage* 3D、Hash* Animation: Master、Autodesk* (旧 Discreet*) 3ds Max* のようなパッケージには、両方のオプションが用意されており、Null オブジェクト (通常はレンダリングできるジオメトリーがない中心またはピボット) やダミー・オブジェクトのようなほかのオブジェクトを通常の骨格階層に統合できます。簡潔に説明するため、この記事ではこれらのオブジェクトをすべて「Null」オブジェクトと呼びます。これらの Null オブジェクトは、キャラクターのメッシュの制御ポイントとともに移動して、スケーリング、回転、移動できます。この手法は、Softimage* 3D で、筋肉の隆起や胸の膨らみの表現に一般に使用されます (図 6 を参照)。

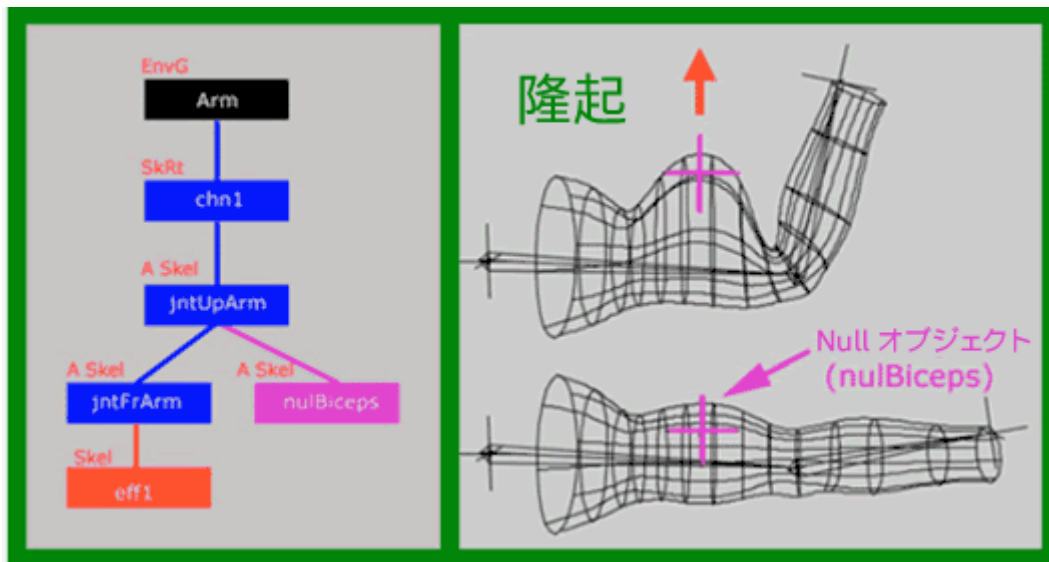


図 6. 骨の階層へ追加した「Null」オブジェクト (紫のオブジェクトおよびボックス) を使用した単純な Softimage* 3D の筋肉隆起のセットアップ。Null オブジェクトの移動、回転、スケーリングは筋肉の位置と隆起に影響し、式により肘の回転と結び付けることができます。図の左の黒のボックスは、このファミリーの親である腕のジオメトリを示しています。青のボックスはそれぞれ、ルートの関節、肘の関節、手の関節で、関節の間の「骨」を回転させます。Softimage* 3D では、関節は選択可能ですが骨は選択できないことに注意してください。Softimage* XSI では、どちらも選択可能です。

個々の骨は、独自のローカル軸を使用して構築されます。通常は、X、Y および Z 軸が使用されますが、常にそうとは限りません。ジオメトリと同様に、これらの軸の配置と向きは骨のピボットポイントと回転方向を決定します。一般に、骨は、骨と (骨が回転できるようにする) 関節の 2 つの部分が接合されたものと考えられます。しかし、すべてのパッケージがこの区別を行うとは限らないことに注意してください。例えば、Softimage* 3D と XSI は、骨セグメント全体を関節と呼んでいます。Maya* や IZWare Mirai* では、関節は接続されていますが、別々に選択可能なオブジェクトです。これにより、関節自体または骨の長さに沿って配置する、「屈折」のようなオブジェクトの変形が可能になります。屈折は、肘の内側 (関節の位置設定に適した候補) の折り目や筋肉の隆起 (骨に沿って位置が設定される屈折) に真実味を持たせることができます。

場合によっては、軸が固定され、ユーザーが骨の中心の向きを変更することはできません。Softimage* 3D では、X 軸は常に骨の長さに沿います。Maya* のようなほかのパッケージは、中心の向きを選択できます。これらのパッケージは、骨の向きに関係なく、中心を自由に回転できます。ただし、やみくもに回転すると、予測できない結果を引き起こすことがあります。

ピボットの向きを決める方法がなぜ重要なのでしょうか。プロのアニメーションでは、処理を正確に行うだけでなく、処理を迅速に行う必要もあります。骨の向きをすべて同じ方法で決めた場合、指、膝、肘、脊椎骨がすべて同じ軸のまわりで前後に曲がることとなります。ほとんどのセットアップでは、その軸は Z です。これにより、式の作成が容易になります。キャラクターが移動している方法を知るために推測したり試行錯誤する必要はありません。

骨を使用した表面変形の例

両手を前に移動して、手のひらを上にします。前腕を見てください。次に、肩を動かさずに、親指が上になるように手を回転させます。前腕を 1 つの固体片として回転した場合、手首と肘の間を捻ることはまず不可能です (仮にできたとしても非常に痛いでしょう)。手に近い肉は手首の関節とともに回転しますが、その効果は肘に近

くなるほど小さくなります。表面の変形では、オーバーレイされたジオメトリの制御ポイントはそれぞれ、最も近い関節または骨により制御されます。

これは、キャラクターがもはや岩のように硬いセグメントを緩める必要がないことを意味します。肘は自然に曲げることができ、上腕二頭筋は収縮できます (図 5 を参照)。

骨の制御

では、最初の時点に戻りましょう。問題を解決するために、完全な骨格と IK システムを含むアプリケーションを購入したとします。問題は解決しましたか? なんとも言えませんね。これは、IK システムは順運動学より直感的であるとは限らないためです。りんごを取ったり歩行中に足を床につけるような動作では、IK は直感的で使いやすく、適切にセットアップを行うことができます。その「セットアップ」は、骨と IK で操作するには多少骨が折れます。セットアップは理解できるようになるまである程度の時間がかかります。

始める場所

自分の関節を見てみましょう。詳細を見る前に、身体がどのように動いているか考えます。CG スケルトンは、人の骨を正確に再現している訳ではなく、あくまでも 1 つの表現であることに注意してください。アニメーション・ツールは限りなく近く現実を反映できるようになっていますが、アニメーションに真実味を持たせることができるかどうかは、アニメーター次第であることに変わりはありません。

さまざまな関節の動きについて考えてみてください。肘と膝の動きは肩や腰の関節の動きとは異なり、互いに少し異なって動きます。また、背中や首の脊椎骨の動きとは全く異なります。一般的に、アニメーション・パッケージでは 1 種類または 2 種類の骨を利用できます。また、骨を制御する、いくつかの異なるオプションも用意されています。人間は、数百の骨で足、脚、手、内臓を支えています。CG スケルトンを構築するときにそれらの骨と内部スケルトンの骨を一致させる必要はまずありません。この手法は、アニメーションの立場から見ると逆効果です。表面を制御するために余分な骨を使用することは、キッチンで余分なコックを働かせることに似ています。妥協して、できるだけ少ない数の骨を使用しましょう。繰り返しますが、ここでの目的は、現実を再現するのではなく、真実味を持たせることです。

関節の種類と制御

真実味を持たせるには、関節部のまわりを (人間が痛みなく動かすことができるよりも) 自由に動かすことができる関節と、人間の肘のように 1 方向にのみ曲がる関節の 2 種類の関節が必要です。Softimage* 3D は、すべての骨格チェーンの最初の (ルート) 関節にすべての軸で自由に回転する能力を与えることにより、球関節に似せています。ユーザーは、(すべての関節を前者のように扱う)「3D」スケルトンまたは (球状のルート関節を除き、人間の肘のように 1 つの平面に沿って 1 つの方向のみに曲がる)「2D」スケルトンのどちらを構築するか選択します。「3D」スケルトンは、首の長い生物や、海草や鎖などを表現するのに適しています。「2D」スケルトン (球関節で始まり肘のような関節で終わる) は、腕や脚を表現するのに適しています。また、一度に 3 つ以上のリンクを使用しない場合にも適しています。

課題への取り組み

一度に 3 つ以上のリンクを使用する必要がある場合はどうすれば良いでしょうか。足のようなものを 3 つ目のリンクとして追加する場合や、IK を使用してりんごを取るキャラクターをアニメーションする場合のような、いくつかの一般的な課題について説明します。

3 目目のリンクのジレンマ: 足を追加するとどうなるか

IK により制御されたチェーンでは 2 つのリンクの処理は簡単でしたが、足が 3 目目のリンクを表すとしたら、どのように対処すれば良いでしょうか。Softimage* 3D では、足首は実際に 2 つのもの (脚チェーンの終点と足チェーンの始点) です。これらの 2 つを組み合わせるには、脚チェーンのエンド・エフェクターの下の足チェーンを親にするか、「コンストレイント」を使用する必要があります。コンストレイントについては後で説明します。まず、2 つのリンクの後にチェーンを IK で制御するのがなぜ難しいのかを説明します。

厳密に言えば、IK により制御されるチェーンは、チェーンの一番下のリンクまたはチェーンの関節により制御されます。下部の関節またはエンド・エフェクターが移動すると、階層の上部までのすべての骨がある程度回転します。

足首が下部の関節でエンド・エフェクターである 2 つの脚の骨 (腿とふくらはぎ) のチェーンを考えます。足首を真っすぐに伸ばすと、膝が曲がり、腿が回転します。これはかなり分かりやすい動きです。では、もう 1 つのリンクとして足の骨を追加してみましょう。同じように (今回は足の親指の付け根のふくらみのエンド・エフェクターで) 伸ばすと、足首が曲がり、膝が曲がり、腿が回転します。しかし、各関節が曲がる順序と各回転の量はわずかな調整でも大きく変化するため、このセットアップを制御することは困難になります (図 7 を参照)。

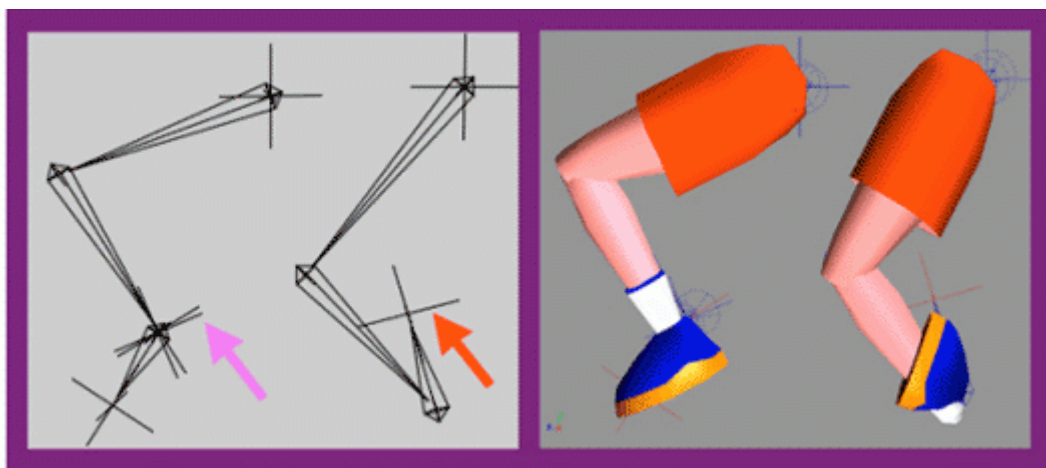


図 7. 多くの骨がある場合、そのまま滑らかに変形できる場合でも、チェーンで 3 つ以上のリンクを使用しないように IK の制御を分割します。左端のセットアップ (紫の矢印) は、Softimage* 3D で 1 つのリンクの足のチェーンを 2 つのリンクの脚のチェーンの親にしたものです。このセットアップにより、足首にエンド・エフェクターが設定され、つま先に別のエンド・エフェクターが設定されます。これは、3 つのリンクのチェーンで 1 つのエンド・エフェクターを使用するよりも適切です。一般に、IK では、骨が少ないことは制御しやすいことを意味します。

2 つの骨では直感的で簡単に制御できていたモーションは、3 つ以上の骨では制御できなくなります。これは、一般的な Softimage* 3D 骨格セットアップで多くのチェーンの親子関係が作成されたりコンストレイントされる 1 つの理由です。ソフトウェアの IK ソリューションで、この点に注目してください。コンストレイントおよび親子関係の作成を創造的に使用する必要があるにもかかわらず、Softimage* 3D は、3D 図アニメーション向けの IK ソリューションを提供しています。

ほかのパッケージは、異なるソリューションを提供しています。Maya* および Mirai* は、骨盤からつま先までの完全なスケルトンを作成することができます (図 8 を参照)。Maya* は、チェーンに配置された個別の「ソルバー」を使用して予測できない長いチェーンを処理します。アニメーターは、ソルバーを必要なときに配置することにより、ソルバーの開始と終了を宣言します。腕を簡単に制御するには、1 目目のソルバーで肩から手首ま

でを操作し、2 つ目のソルバーで手首から (手の) 中指の付け根までを操作します。手と腕のアニメーションには、各ソルバーの別々の操作が含まれます。

Maya* には、首の長い生物をアニメーションする特別な「スプライン」ソルバーも用意されています。このソルバーを使用すると、首の脊椎骨にカーブ (スプライン) が描かれます。カーブを操作すると、脊椎骨が回転してカーブの形になります。海洋生物を描くための非常に優れたソリューションです。

IZWare のアニメーション・ソフトウェアは、ディズニーのトロン以降、多くの映画で利用されています。多角形のモデル化パッケージで有名です。新しい「既製の」アニメーション・パッケージである IZWare Mirai* は、アニメーターが瞬時にキャラクターを作成してアニメーションできるように支援します。これが可能なのは、IZWare が前もって多くの作業を行っているためです。Mirai* は、Maya* のように完全な分岐したスケルトンを作成できるだけでなく、さまざまなキャラクター (人間、犬、ダニなど) を完璧に表現する既製のスケルトンも提供します。



図 8. Mirai* の既製モデルとスケルトンで表現された少年、犬、そして大きなダニ

自動ミラーリングと逆回転問題の解決

IZWare は、手足の動きを自動的にミラーリング (反転) する多くの表現とキャラクター制御を追加することにより、テクニカル・ディレクターの頭を悩ませていた問題を解決しました。これらの動きの例には、歩くまたは走るときに手と足を反対 (例えば、手は前、足は後ろ) に動かして腕を振り上げる動きが含まれます。手足の 1 つをアニメーションした後、逆の手足を自動的にアニメーションするかどうかを選択する必要があります。これにより、半分のスケルトンを作成してほかのプログラムにミラーコピーしたときに生じる逆回転問題も修正されます。(例えば、ミラーコピーされた関節の中心も「ミラー」されるため、関節は当然逆方向に回転します。)

IK のコンストレイントへの取り組み

スケルトンを作成し、IK を使用して足を制御して床の上に立たせました。手にも IK を使用しました。1 歩進んだところで、腕がおかしいことに気がきました (図 9 を参照)。

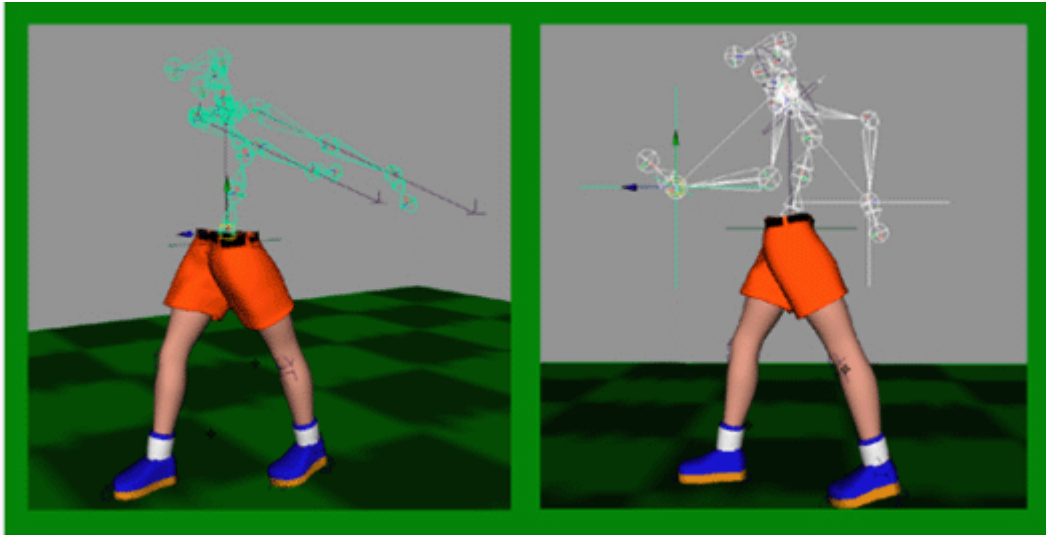


図 9. IK のエンド・エフェクターの位置は、(単純な階層のように) 親の場所に関連するのではなく、空間でキーフレームされます。IK で足の配置は非常に簡単になりましたが、スケルトンに関連させる必要がある手首の位置設定は適切に行われていません。簡単な修正方法は、手首のエンド・エフェクターを Null オブジェクトまたはロケーター (右の図の十字線) にコンストレイント (制約) した後、それらのロケーターを階層の親にすることです。

ここでコンストレイントの出番です。コンストレイントにより、エンド・エフェクターをキーフレームにしてもうまくいくことはめったにないことが判明しました。実際にそうすることで、IK で見つかるほとんどの問題が発生し、キャラクターが絵コンテで注意深くプロットした巧みな動きの多くを行うのを防ぐことができます。

コンストレイントは、自由自在にオンオフを切り替えることができる強力な磁力と考えると良いでしょう。1 つのオブジェクト (Null オブジェクトを含む) がほかのオブジェクトに影響を与えることができます。異なるコンストレイントは、異なる処理を行います。コンストレイントは、ほとんどの場合、オブジェクトの中心またはピボットに向けて作用するため、オブジェクトの中心またはピボットポイントの配置が非常に重要になります。一部のパッケージでは、ほかのパッケージよりも多くの種類のコンストレイントが用意されていますが、次の 3 つの基本的なコンストレイントが定番です。

- 目標または方向
- 向き
- ポイントまたは位置

目標または方向

目標または方向のコンストレイントは、「影響される」オブジェクトを「ターゲット」オブジェクトの中心またはピボットの軸 (一部のパッケージでは選択できます) に向けます。例えば、ウィンブルドンの観客の顔は、常にバウンドしてネットの上を行き来するテニスボールを追います。

向き

向きのコンストレイントはシンクロナイズの選手に似ています。1 人の選手が振り向くと、もう 1 人の選手も振り向きま。影響されるオブジェクトの中心の向きはターゲット・オブジェクトの向きと一致します。

ポイントまたは位置

ポイントまたは位置コンストレイントは、ほとんどの場合 IK とともに使用されます。このコンストレイントは、各オブジェクトの中心位置に基づいて、1 つのオブジェクトを別のオブジェクトの上に直接固定します。1 つのオブジェクトが移動すると、別のオブジェクトも移動します。

コンストレイントのターゲットとして使用するオブジェクトは、ほかのオブジェクトの親になります。つまり、オブジェクトは基本的な階層の規則に従うことを意味します。階層では、オブジェクトの位置と動きは親に関連しています。

りんごを取る問題

図 9 でスケルトンの手が後ろで固定されているのは、コンストレイントのためです。これは、骨格階層と基本的なオブジェクトまたはジオメトリー階層の間の主な違いの 1 つです。IK アニメーションは通常、両方の種類を含んでいます。階層のオブジェクトの例として、編隊を組んで飛行している飛行機の前頭の飛行機を親として考えます。そのうちの 1 機をグループの上または下でロールさせると決めた場合、その動きをキーフレームにすると、その 1 機はグループのまわりをロールしながら、グループとともに飛び続けます。これは、この 1 機の動きが親オブジェクトである前頭の飛行機の動きに関連しているためです。

基本的な階層では、子の位置はその子の親の位置に関連しているため、階層 (骨なし) で結合されたキャラクターが腰を回転すると足が動きます。

逆運動学チェーンでは違います。エンド・エフェクター (IK 階層の最も下部の子) の位置は空間に関連しているため、IK でアニメーションされた足は動きません。IK でアニメーションされた手は、キーフレームされた空間のポイントに移動しようとしています。

IK は足の配置に最適でした。では、りんごを取るキャラクターはどうでしょう。IK を使用すると、その動作はより簡単になるでしょうか。もちろんです。その動作も、基本的な階層とコンストレイントを利用して行うべきです。

肩、上腕、前腕を順に回転するよりも、りんごに手を置くほうがはるかに簡単であるため、IK を使用したいところです。しかし、キャラクターの手が後ろの空間を引きずりながらキャラクターがりんごの木まで歩くのも困ります。どうすれば良いのでしょうか。

りんごを取るソリューション

IK ポイント・コンストレイントを基本的な階層とともに使用します。

- 2 つのオブジェクト (正方形と円) を使用します。これらはスプラインのみでレンダリングしないため、後で非表示にする必要はありません。ここでは円を使用します。
- 1 つの手首 (腕チェーンのエンド・エフェクター) に 1 つの円を配置し、別の手首に別の円を配置します。ポイント・コンストレイントを使用して手首に円を直接置いた後、それらのコンストレイントをオフにします。
- それらの円をキャラクターの胴体の子にします (胴体を親にします)。これは、胴体が移動するとそれらの円も移動し、円のすべての動きは胴体に関連することを意味します。それぞれの円に対して手首のエンド・エフェクターを抑制します。ポイントは、エンド・エフェクターではなくコンストレイント・オブジェクトをキーフレームにすることです。

最終結果は、両方の長所を兼ね備えたものとなりました。キャラクターが歩くと、手は胴体とともに移動します。りんごの上にコンストレイント円を置くと、IK アニメーションにより、手はその上に移動します。

コンストレイント・オブジェクトの階層を使用して IK スケルトンを操作すると、キャラクター・アニメーションの問題の約 95 パーセントは解決するでしょう。馬から落ちている騎手と馬へ飛び乗っている騎手の 2 人のキャラクターはすべて、アニメーションされたコンストレイント階層と IK で表現することができます。足の回転のような複雑な動作を制御する式に小さな計算を追加することにより、あらゆるものの制御に必要なツールを手に入れることができます。

まとめ

ロボットの電源を切り、キャラクターに単一メッシュの鎧を着せたり表面を変形するのは止めて、新しい手法を利用しましょう。この記事では、階層、逆運動学と順運動学、それらに対応した主なアニメーション・ソフトウェア・パッケージについて説明しました。また、IK とコンストレイントを使用して、いくつかの一般的な (対応が難しい) アニメーションの問題を解決しました。

自分で試すかどうかは皆さん次第です。最初にエンジンが IK とコンストレイント階層をサポートしていることを確認し、次にそれらを利用してください。使用している 3D ソフトウェア・パッケージがこの記事で紹介されていない場合でも、ほとんどのパッケージは、IK システム、階層、何らかのコンストレイントまたはリンクをサポートしています。価格は関係ありません。利用できる機能を調べて、それらを活用し、素晴らしいアニメーションを作成してください。

参考資料

この記事で説明したツールおよびソフトウェア・パッケージの URL を以下にリストします。インテル® ソフトウェア開発ツールの一覧は、<https://www.isus.jp/products-list/> を参照してください。

- Maya* は Alias* PowerAnimator* および Wavefront* Explorer の後継ソフトウェアで、現在は (旧 Discreet*) 3ds Max* とともに [AutoDesk*](#) (英語) が所有しています。
- [Hash* Animation: Master](#) (英語)
- インテル® Pentium® III プロセッサでアニメーションをスムーズに行う (NURBS モデル)
- [Softimage* 3D](#) および [XSI](#) (英語)

コンパイラーの最適化に関する詳細は、[最適化に関する注意事項](#)を参照してください。