空間的関係の記述子を用いた アニメーションの作成*

Spatial Relationship Descriptors to Synthesize Animations

幸村 琢**
Taku KOMURA

Key words character animation, spatial relationship, character interaction

1. 序 章

キャラクターが複雑に絡み合う、紐同士が結ばれる、布で物を包むといったアニメーションを生成するのは大変手間がかかる上に難しい。物体そのものの自由度が高く、しかも衝突が常に起こるため、相互依存性を考慮に入れながら自由度を減らす必要が出てくる。その際に空間的関係の記述が大変重要な意味をもってくる。本解説ではこれまで提案されてきた空間的関係に基づく記述子について説明する。特に Intareaction Mesh, Gauss Linking Numbers, Geodesic Control, 流束を用いたアニメーション生成について述べる。また、今後の課題についても議論する。

アニメーションを生成する作業は非常に時間がかかる。キャラクターアニメーションの場合、ボーン構造の姿勢を指定するたくさんのキーフレームを生成し、関節角度を補間しなければいけない。キャラクターが他のキャラクターと相互に干渉するような動作の場合はさらに複雑度が増すものである。人の動きに応じてもう一人のキャラクターも反応しなければ自然な動作にならない上、体同士が簡単に衝突、貫通してしまう場合が多いのでより慎重にキーフレームを生成する必要があるのである。

根本的にもっとも大きな問題だと思われるのは状態空間の記述の仕方である。人間の状態の記述はよく関節角度を用いて記述される。この記述手法はロボット工学から来ているが、階層上位のセグメントに対しての相対的な平行移動、回転量に基づいており、根となるセグメントから木構造上に葉に向かって広がり、キャラクターの姿勢を決定する。体の衝突状態、接触状態は状態空間の要素としては記述されていない。

しかし、われわれが日常生活している際、もっとも気にしているのは自分と他の人や環境との空間的な関係ではないだろうか。自分が例えば席に座ってパソコンで仕事をしているのを定義しているのは自分の関節角度よりも尻が座

席部分に接触し、顔がパソコンを向き、指でキーボードをたたいている、空間的関係によって決定づけられている。体の大きい人は多少体を曲げるだろうし、小さい人は逆にモニターを見上げる状態になるかもしれない。よって関節角度は多少違いがでてくる。ただし、最初にした関係は同じである。アニメーションにおいてもやはりそうなるはずである。キャラクターが一人で何かをしているシーンはあまりなく、部屋の中で他のキャラクターと話したり格闘したりしているシーンが多い。よって独立した関節角度による状態空間ではなく、それぞれの要素がどのように干渉しているかに基づく状態空間があればキャラクターの操作もやりやすくなり、またアニメーションの再利用もしやすくなる。

この相対的な関係を表す記述による操作はロボットの制御等においても重要である。ロボットに人間の真似をさせる“learning by demonstration”という発想があるが、まさにこの関係を用いた記述が大事である。ロボットの体の自由度は人間の自由度よりも多くの場合低い。また関節構造も体の位相も違う。関節角度はそのためほとんど意味をなさない。大事になってくるのは体の部分同士あるいは体と環境の空間的な関係である。実際に人間はそのように他の人の動作を認識しているのは明らかである。子供は体のサイズの違いがあるにもかかわらず、大人の動きを見て瞬時に真似ができるし、ロボットやアニメのキャラクターの動作を見たらすぐに真似ができる。ロボットに人間の真似をさせるためには空間的関係を用いた動作の記述は不可欠である。

同じことは紐や布のアニメーションでもいえる。多くの場合、紐や布の状態はそれを構成するパーティクルの位置や速度によって記述される。もちろん、シミュレーションのためにはそうする以外にはないのだが、人間がその状態を認識したり、あるいはそれらの物体を操作するには抽象化が必要になってくる。その際に重要になってくるのはそれぞれの布の部分的、あるいは全体的な空間的関係である。紐が硬結びされているか蝶々結びされているか、あるいは布が箱を包んでいるか箱の中に入っているか、等、

*原稿受付 平成 25 年 3 月 20 日

**School of Informatics, University of Edinburgh (10 Crichton Street, Edinburgh, UK, EH8 9AB)

われわれの認識、生活において大事とされるのは紐や布のパーティクルの絶対位置よりも、それらが他の物体やそれ自身という空間的関係を構成しているかである。

本解説ではアニメーションの分野において用いられてきたそのような空間的な関係に基づく記述子について述べる。まず、最初にこれまで用いられてきた空間的記述子について説明した後、より抽象化された記述子について解説する。最後に今後の展開について解説する。

2. キャラクター同士あるいは物体との関係の記述方法

まずはより自由度の低い人間型キャラクターを例に挙げて実用的な手法を説明する。ここでは実際にキャラクターが物を運んでいる状態、あるいはキャラクター同士がレスリング等をしている状態を記述する方法を考えてみる。具体的な手法として接触情報、距離情報、ラプラシアン座標を用いた手法についてそれぞれ説明する。

2.1 接触の使用

最初に考えられるのは接触状態を離散的に指定することによりキャラクター同士の関係、あるいはキャラクターと物体の関係を記述する方法である。もちろん、それだけでは冗長性が高すぎるため、状態の記述には不十分である。それで関節角度と組み合わせることによって記述する。これが実はもっとも一般的な記述の仕方である。物を掴んだりする動作を生成するときには手の軌道と握る物体の軌道を同じにすることによって手と物体が同期して動く。問題が起きるのは手が徐々に近づいてきて接触を開始するときである。接触があるかないかは離散的な状態として記述されているため、不連続な姿勢を作らないためには接触が開始されるフレームには手が必ず対象のある場所になければならない。この条件を満たすためには二つの方法がある。動作全体を入力としてそのような拘束条件を満たすように動作全体を同時に最適化する spacetime constraints という最適化手法を用いるか¹⁾、体と対象の距離に応じてその重み付けを変化させる手法を用いるかである²⁾。前者はオフラインなアプローチなので、アニメーション編集に向いているといえる。後者はフレームごとに動作を計算するのでリアルタイムにキャラクターを動かすゲーム等に向いている。しかしどちらにしろ、はっきりと接触がどこで起こるかがわかっている場合はよいがそうでない場合、例えば体全体で物を囲い込むような、接触が安定に起きていない動作や体をスライドさせるランバダのような踊りを扱うときに問題が出てくる。また、接触する部分以外の部分の関係の記述がない場合、体が物体に衝突したりすることが頻繁に起きてしまう。そうした場合、衝突判定を行い、食い込みが起きている場所を解消しなければならない。特に絡み合うような動作の場合はこういうことが頻繁に起き、また凹物体を扱う場合に体が場所によって違う方向に押し返されることがあるので、動作が不安定になる。したがって、接触により空間的関係を説明する手法は非常によく用いられているが、複雑な動作を扱う際には問題が起こって

図1 体の間の空間に定義されたインタラクションメッシュ

しまう。

2.2 距離情報による記述

次に体同士が近い部位を指定し、その距離間のユークリッド距離を与えることにより、状態を記述することが考えられる。それらの部位の間の距離の時系列データより、その動作が特徴付けられる。実際に文献3)では物体間の距離の関係を Minimum Spanning Tree によって記述し、一つのシーンの空間的関係データを他のシーンに適用している。簡単な例としては体と服の間の距離の関係を記述し、姿勢が変わった際に体と服の関係が保持されることにより、服が体の表面から一定の距離を保つようにしている。ただしこの手法では物体間の関係が時系列に変化することは想定していない。

2.3 Laplacian 座標を用いた手法

文献4)では体間の関係を Laplacian 座標を使って表現する手法を提案している。すなわち、体の関節の位置を点群としてドローネ四面体分割を行い、体と体、あるいは体と環境の間の空間を表すボリュームグラフ(図1参照)を生成し、空間的関係を表現している。このボリュームグラフは体と体、あるいは体と環境の間の関係を表しているのでここでは特に Interaction Mesh と呼んでいる。Interaction Mesh の局所的な形状は体の大きさや姿勢が多少変化しても一定であるべきであるという仮定の下、動作を他のサイズのキャラクターに適用したり(リターゲットイング)やシーンの編集を行っている。Interaction Mesh の状態を表す手法としてコンピュータグラフィックスで形状編集を行うのによく用いられる Laplacian 座標を用いている。ボリュームグラフを使った物体の変形を行った研究は以前に提案されているが⁵⁾、この研究では開空間においてそれを利用している。この手法の特徴は、キャラクターの動作やサイズが変更されて体が他の部分や環境と衝突しそうな場合でも開空間の形状を保つ作用により、体の姿勢が変形され、衝突を避ける効果が得られる。

この手法は文献3)が対応していなかったダイナミックに関係が変化する場合にも対応している。フレームごとに Interaction Mesh が計算しなおされ、体や環境のサイズを

図2 インタクションメッシュを用いた柔道動作のリターゲティング

図3 手を回す動作(中)を腕の長いキャラクターに関節角を用いてリターゲティングした場合(左)とインタクションメッシュを用いてリターゲティングした例(右)

変えた際にシーンの連続性が保たれるよう関節に対応するパーティクルの加速度の二乗和を抑制することによりこれを行っている。スムーズさを保つためにすべてのフレームにおいて一括的に計算を行っているため、リアルタイムの編集には向いていない。

図2にこの手法を用いて動作をリターゲティングした例を示す。入力として柔道をしている二人のキャラクターの動作があった場合(左から3番目のペア)、体のサイズを変えても同様な動作が実行されることを示している。既存の手法では体同士の関係を接触や距離を用いて記述するため、このように複雑に絡まりあった動作を扱うことができないが、この手法ではスムーズな動作が生成されている。一人のキャラクターが手を前後方向に入れ替えるように前腕を含めてくるくる回す動作(図3参照)を腕の長いキャラクターに変更した場合、既存手法では手が衝突して貫通したりするが、この手法では腕同士の空間的關係が保たれているため、元の動作のコンテキストが保たれている。図4に示す例では、入力としてキャラクターが車に乗り込む動作があった場合、車のサイズが変更されても同様にうまく乗り込むことができるように動作が自動的に変更されている。注目されるべきところは足が車の障害物をよけながら乗り込んでいるところである。

3. 空間的関係の抽象化

このように体同士の関係を用いた記述子は有効であることがわかるが、動作そのものの情報量は実は高くなってしまいう場合が多い。というのは状態を示すのに部分同士のペアを記述子にしているためである。ここではより抽象的な人体動作の表現を用いた空間的関係の方法を説明する。

図4 車に乗り込む動作のリターゲティング

まずは人間の体自体の抽象化を行う必要がある。人間には太っている、痩せている、大きい、小さいなどさまざまな身体的特徴をもっているが、単純には頭、両足、両腕、胴体で構成され、図5で示されるような一次元複体のスティックフィギュアとして考えることができる。次に二人のキャラクターがダンスやプロレスの技などなんでもよいが近くで絡まっている姿勢をとっている状態を考える。このように表現された二つの姿勢がどれだけ似ているか(姿勢間の距離)はどのように定義することができるだろうか。この距離の定義は抽象化において大事である。というのは距離を定義することにより、動作が似ているかどうか決定され、さらに動作の分類を行うことができるからである。しかし例えば関節角度の差の二乗和などを使ってはいけない。なぜなら例えば図5に示すような二つの姿勢があった場合、これらを同じ姿勢であると判断してしまい、関節角度空間で線形に補間してしまうと、足や腕がお互いを貫通してしまうからである。図5で示されている二つの姿勢は関節角度空間で近くても実際の動作空間では遠い姿勢なのである。

このように定義された体の姿勢の距離を決めるのは絡み目である。絡み目は二本の紐がお互いに重なりあっている

図5 関節空間では似ていても実際には二つの姿勢の違いは大きい

図6 さまざまな結び目

図7 投影面によって紐の状態が変化してしまう

様を表しており（図6参照）、結び目理論で結び目の一部分を定義するのに用いられる。さまざまな種類の結び目は結び目の組み合わせにより生成することができる。ある結び目と他の結び目の間の距離（すなわちどれだけ紐を解きまた絡ませるべきか）は限られた場合に計算することができるが、人間の姿勢の場合、複雑な結び目を作ることは不可能なのでどれだけ姿勢が似ているかは基本的に各四肢のペアがどれだけ絡んでいるかを比べることによって計算することができる⁶⁾。

しかし最後に残る問題としては、三次元的に絡み合っている体同士の絡み数をどのように求めるかである。そもそも結び目理論は結び目を表すのに二次元の平面にそれを投影し、どのように紐が重なっているかによって状態を決定している。開いた紐の場合、投影面の方向により、重なり具合が変わってしまう（図7参照）。この問題は Gauss Linking Integral を用いることによって解決することができる。リンク数（Linking Number）は二つの閉じた紐がお互いを何度回っているかを示す位相的な数である。ガウスは以下の式（Gauss Linking Integral）を用いてリンク数が計算できることを示した。

$$GLI(\gamma_1, \gamma_2) = \frac{1}{4\pi} \int_{\gamma_1} \int_{\gamma_2} \frac{d\gamma_1 \times d\gamma_2 \cdot (\gamma_1 - \gamma_2)}{\|\gamma_1 - \gamma_2\|^3}$$

ここで γ_1, γ_2 は二つの紐のそれぞれの三次元軌道を表している（ガウスがどうやって導出したかは文献7）を参照）。

この式のすばらしいところは値が開いた紐でも計算で

図8 さまざまな状態の紐の Gauss Linking Number

き、閉じた状態と開いた状態で値が連続的に変わるところである。三次元的に直接計算できるため、二次元に投影する必要もない。積分を行い、その出力値（Gauss Linking Number）を見ることによって、どれだけ二つのロープが絡み合っているかを推定することができる。例えば図8に示すように絡み具合によって出力される値が変化する。面白いことに値は符号をもっていて、順周りと逆周りで符号が変わってくる。

さらに開いた状態の紐でこの値を計算し、その値が増加する方向に紐を導くことにより、紐を絡ませることができる。つまりは拘束条件としてこの Gauss Linking Number を用いて紐の動作を導く。また、紐が人間の手足を表している場合、その値を制御することにより、レスリングやダンスなど手足が絡まるような動作を誘導できる⁸⁾。またもう一つ面白いところは紐がお互いを貫通すると値が不連続に変化することである。これはすなわち、Gauss Linking Integral が紐の状態を表す多様体上の関数であることがいえる。つまり、その値を逐次検査することにより、不当な動作が起きたかどうかをチェックできる。

体のような複体に対応するため、体の部分を結ぶ線を定義し、それぞれについて Gauss Linking Number を指定することによって、いろいろな絡み合う動作を生成した例を図9に示す。ここではそれぞれの体の関係が抽象的な Gauss Linking Number で定義されているため、前方のキャラクターの動作に簡単に対応できているところが見て取れる。これはまさに抽象化のもたらすメリットである。

上では二本の紐についての Gauss Linking Number について述べたが、実は一本の紐だけでも計算することができる。二重積分の紐を同じ紐に対して重複して計算することによって紐が自分自身とどれだけ絡んでいるかを計算できる⁹⁾。

ただし、ここで問題となってくるのは細かい操作である。紐を絡ませるときに右から左に向けて絡ませるのかそれとも逆に絡ませるのか、ゆるく絡ませるのか、それともきつく絡ませるのか、といった細かい特徴量を指定できないと実は操作が困難である。Gauss Linking Number を拘束条件として与えた後、例えば関節角の変化の二乗和を最小にしながら Gauss Linking Number を増やすように動かすと紐が遠くから大きく回るように動き、われわれが普段紐を動かすのとは違う動きになる。われわれが紐等进行操作するときには実際にはどうするかを思い返してみると、図

図9 絡み合う動作の生成

図10 Geodesic Control による紐の巻きつけ

図11 障害物がある中での Geodesic Control

図12 普通の包み方

10のようにまず紐をクロスさせ、次に片方の紐をもう一本の紐に巻きつけながら接触面を徐々に増やしていくの気づく。実はこの操作方法は非常に効率的でなおかつ紐の最小限の自由度を用いて行えることがわかる。例えば右向き、左向きと巻き方を変えたいなら最初にクロスさせるときの角度を90度より多少ふやさか減らすかによって向きを調整できる。また紐と紐の接触面を徐々に増やしていくだけであるため、操作に冗長性がなく、たとえ、少々障害物があっても（指が入ることができれば）うまく絡ませることができる（図11参照）。巻きつけられる紐の周りに Geodesic Line を書き、それにそってもう一つの紐を巻きつけるので、この手法を Geodesic Control と呼んでいる¹⁰⁾。

Geodesic Control の趣旨は多自由度の物体の操作を簡単

図13 ワインボトル包み

図14 いろいろな風呂敷包み

図15 ガウスの法則の概念図

にするということであり、このアイディアは布を何かに巻きつけるときにも応用可能である。すなわち物を布に載せた場合、多くの場合、われわれは布をまずびんと張った後、徐々に布と物体の接触面を増やすように包む。こうすることによって、布の冗長性がなくなり、操作も簡略化される。図12～14に Geodesic Control を利用して生成したさまざまな風呂敷包みの例を示す。

最後に物を包む際の包まれる物体と包む布の空間的な関係を示す量について考える。Gauss Linking Number が紐同士の絡み合いを定義したように物が布にどれだけ包まれているかを示す量があれば、バッグに物を入れたり、皮を

図 16 流束を増やすことによる袋の制御

図 17 流束を増やすことにより生成されたズボンをはく動作

むいたりといった操作に使えるかもしれない。

そこでまた物理書を見るとガウスの法則というのがある。ガウスの法則によると電荷から出る流束 (Flux) は包み方の幾何的形狀によらず一定である (図 15 参照)。流束は以下の積分式で求めることができる。

$$\Phi = \oint_S E \cdot dA = \oint_S E \cdot \hat{n} dA = \frac{Q}{\epsilon_0} = \text{const.}$$

ここでは E は電界を表し、 S は電荷を帯びた物体を包む任意のサーフェスであり、 dA は S の極小面積 ($\hat{n}$ はその法線ベクトル)、 Q は物体の電荷で ϵ_0 は電気定数である。ここで思いつくのは流束を用いて物体を包む操作を行うことである。物体に仮に帯電させ、それにより作り出される電界の流束を増やすように布をコントロールしたらどうなるかということで実験した結果が図 16, 17 に示されている¹¹⁾。予想通り、布は物体を包む方向に動いてゆき、包み終わったところで動作が収束する。流束のよいところはそれが物体の幾何的形狀や凹凸、さらには位相によらず連続的な量であることである。よって流束を増やす方向に布を動かしてゆけば布の伸縮に問題がなければ必ず正しく包めるのである。

4. 結 べ り に

本稿では物体の空間的関係の記述子とその応用について簡単に述べた。空間的関係は複雑な捜査を行うときに必需品であるといえる。機械学習等で多関節物体、紐や布といった冗長性の高い物体をコントロールをする際にも非常に重要になってくる。今後は連続性の高く、また多重解像度の性質をもつ空間的関係の記述子の開発、また空間的関係の記述子を用いたモーシヨンプランニング等が重要になるであろう。

参 考 文 献

- 1) M. Gleicher: Motion Editing with Spacetime Constraints,

- Proceedings of the 1997 Symposium on Interactive 3D Graphics.
- 2) H.J. Shin, J. Lee, S.Y. Shin nad M. Gleicher: Computer Puppetry: An Importance-Based Approach, ACM Transactions on Graphics, **20**, 2 (2002) 67-94.
- 3) K. Zhou, W. Xu, Y. Tong and M. Desbrun: Deformation Transfer to Multi-component Objects, Computer Graphics Forum, **29**, 2 (2010).
- 4) E.S.L. Ho, T. Komura and C.-L. Tai: Spatial Relationship Preserving Character Motion Adaptation, ACM Transactions on Graphics, **29**, 4 (2010), (Proceedings of SIGGRAPH 2010).
- 5) K. Zhou, J. Huang, J. Snyder, X. Liu, H. Bao, B. Guo and H.Y. Shum: Large Mesh Deformation Using the Volumetric Graph Laplacian, ACM Transactions on Graphics, **24**, 3 (2005) 496-503.
- 6) E.S.L. Ho and T. Komura: Indexing and Retrieving Motions of Characters in Close Contact, IEEE Transactions on Visualization and Computer Graphics, **15**, 3 (2009) 481-492.
- 7) R.L. Ricca and B. Nipoti: Gauss Linking Number Revisited, Journal of Knot Theory and Its Ramifications, **20**, 10 (2011) 1325-1343.
- 8) E.S.L. Ho and T. Komura: Character Motion Synthesis by Topology Coordinates, Computer Graphics Forum (Eurographics 2009), **28**, 2 (2009).
- 9) P. Røgen and B. Fain: Automatic Classification of Protein Structure by Using Gauss Integrals, Proceedings of the National Academy of Sciences, **100**, 1 (2003) 119-124.
- 10) H. Wang and T. Komura: Manipulation of Flexible Objects by Geodesic Control, Computer Graphics Forum, **31**, 2 (2012), (Proceedings of Eurographics 2012).
- 11) H. Wang: Control of Objects with a High Degree of Freedom, PhD Thesis 2012, University of Edinburgh.

幸村 琢

1995 年東京大学理学部情報科学卒。2000 年東京大学理学部情報科学専攻博士。理化学研究所研究員、香港城市大学講師を経て、2006 年からエジンバラ大学情報学部講師。現在准教授。研究内容はコンピューターグラフィックス、キャラクターアニメーション、ロボティクス。