

J_019

多関節弾性物体モデルのための計算コストと使用メモリ削減法の検討

Study of Reducing Computation Costs and Memory Usage for Articulated Elastic Object Models

渡辺 隆史*1 大谷 淳*1 棚沢 順*2,*1 徳永 幸生*3
Takafumi WATANABE Jun OHYA Jun KURUMISAWA Yukio TOKUNAGA

1. まえがき

筆者らは局所的な変形を含む伸縮運動を可能にした、多関節弾性物体モデルを検討している[1]。本モデルの応用としては、イモムシやたこなどの軟体動物やキャラクターモデル（例えば、テレビアニメーションなどに登場する動物や植物を擬人化したキャラクターの弾性物体モデル）への適用が挙げられる。

筆者らの多関節弾性物体モデルは、頂点数 2000 程度の物体に対して、物理シミュレーションを用いてリアルタイムで変形が可能であった。しかし、問題点として、①計算コストが頂点数の 2 乗に比例するため、計算コストが大きくなりやすい。②変形の計算には大きな行列が必要となり、多くのメモリが必要となる。③弾性物体の変形の際に反復計算が必要となるため、数値の発散が発生してしまう、ことが挙げられる。これらの問題は、境界要素法などを用いた弾性物体の変形計算に起因する。

前述のような問題を解決するため、本論文では、変形の計算に用いる境界要素法の効率的な計算手法およびそれに適した多関節弾性物体モデルを提案する。境界要素法を用いる場合、弾性物体の変形は行列の計算となる。従って、行列を小さくすることで、計算が速くなり、使用メモリが減らせる。また、以前のモデルでは、境界要素法で求まる表面力を用いて境界条件を設定していたが、これが反復計算および計算コストの増大の原因となるため、表面力は用いないようにする。

本論文では実験により、従来モデルの 10 倍以上の計算速度を得られること、および、反復計算をせずに伸縮の変形が可能なることを示す。

2. 多関節弾性物体の構造

図 1 は、多関節弾性物体の構造を示す[1]。

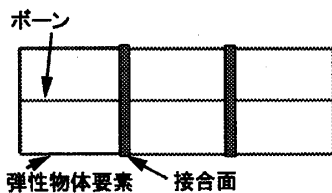


図 1 多関節弾性物体の構造

多関節弾性物体は、図 1 のように弾性物体要素とボーンと接合面からなる。そして、ユーザがボーンの端点を操作することで、多関節弾性物体を伸縮運動させる。伸縮運動は次の手順で行う。

はじめに、ボーンをインバースキネマティクスによって計算する。次にボーンの伸縮のアニメーションを行う。ボーンアニメーションは、各関節点に対して、両隣のボーンの長さおよびその間の角度に依存した力を利用し、運動方程式を立てることで行う[1]。次に、弾性物体要素をボーンに取り付けていく。弾性物体要素は単一の弾性物体とし、変形には境界要素法を用いる。そのとき、弾性物体要素は接合面に接合するように境界条件を与え、変形する。最後に、接合面の各頂点を、境界要素法によって求めた各頂点にはたらく力から運動方程式を立てることでアニメーションする。ただし、提案モデルでは接合面のアニメーションは行わない。

3. 計算コスト削減のための境界要素法

多関節弾性物体の計算の中で、境界要素法を効率的に計算することで、モデル全体の計算の効率化が図れる。

弾性物体の境界要素法の計算は式 (1) となる[2]。

$$HU = GT \quad (1)$$

ここで、 U, T はそれぞれ、弾性物体の各頂点における変位および表面力の値を並べた $n \times 1$ の行列 (n は、弾性物体の頂点数 \times 次元)、 H, U は境界要素法によって得られた $n \times n$ の行列である。各頂点では、変位または表面力の一方を未知の値、もう一方を既知の値とする。既知の値を右辺、未知の値を左辺に移項すれば連立方程式の問題に帰着する。

$$AX = BZ \quad (2)$$

X, Z はそれぞれ、 U または T の未知の値および既知の値からなる $n \times 1$ の行列、 A, B は、 H または G の成分からなる $n \times n$ の行列である。式 (2) の解は、式 (3) となる。

$$X = (A^{-1}B)Z \quad (3)$$

したがって、計算コストは $n \times n$ の行列と $n \times 1$ の行列の積となる。ただし、 Z の成分は、接合面に対応する頂点以外では 0 となるため、接合面の頂点数 \times 次元を m とすれば、計算コストはおおよそ $n \times m$ に比例することになる。さらに、次の変形を考える。

$$X = A^{-1}BCv \equiv A'v \quad (4)$$

ここで、 $Z=Cv$ (C : 定数、 v : 変数) とした。このとき、 A' が $n \times g$ ($g < m$) の行列となれば、計算コストは $n \times g$ に比例することになる。したがって、この操作によって、計算コストの削減が可能であり、行列も小さくなるために使用メモリも減らすことができる。また、提案手法では表面力の解を求める必要が無いので、 X の行数は接合面以外の頂点数 \times 次元 n' ($< n$) とでき、最終的な計算コストは $n' \times g$ となる。

$Z=Cv$ の式は、接合面の変位、つまりは、接合面の形状をパラメータ v で表すことができれば良いことを示してい

*1 早稲田大学 Waseda University
*2 千葉商科大学 Chiba University of commerce
*3 芝浦工業大学 Shibaura Institute of Technology

る。本論文ではこの Cv を、式 (5) を用いて計算する。図 2 は、式 (5) の各種値を示している。

$$Cv = \begin{pmatrix} x & y & 0 & 0 & 0 & 0 \\ 0 & x & y & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 \end{pmatrix} \begin{pmatrix} C_1^2 C_2 & S_1^2 \\ S_1 C_1 C_2 & -S_1 C_1 \\ S_1^2 C_2 & C_1^2 \\ C_1 S_2 & 0 \\ S_1 S_2 & 0 \\ dz/a & 0 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix}$$

ただし、 $C_1 = \cos \theta_1, C_2 = \cos \theta_2, S_1 = \sin \theta_1, S_2 = \sin \theta_2$
(5)

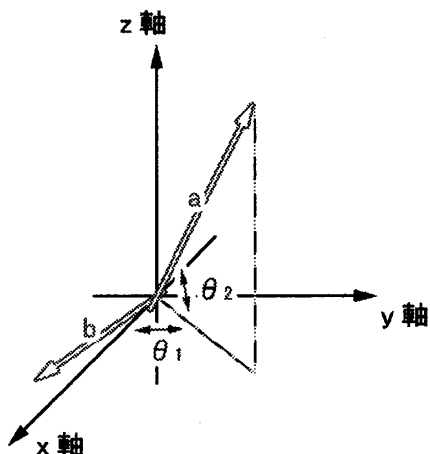


図2 接合面の計算

式 (5) は、接合面が図 2 の x-y 平面にある場合に、接合面を任意の軸方向 (図 2 の a 軸)、および、その軸と垂直な方向 (図 2 の b 軸) が作る平面上に回転し、a 軸および b 軸方向に拡大縮小できる場合の Cv である。式 (5) において、 x, y, z は弾性物体要素の接合面部の頂点の座標である。a 軸は、x 軸と θ_1 、x-y 平面と θ_2 の角度を持っているとする。また、b 軸は、x-y 平面上にあり、a 軸と垂直である。式 (5) の a, b は、a 軸および b 軸方向の拡大率である。dz は、z 方向への接合面の平行移動量とする。

この場合、弾性物体要素の形状は a, b, θ_1, θ_2, dz の 5 つのパラメータで決定される。このうち、 θ_1, θ_2, dz の 3 つのパラメータは、ボーン長さや角度から決定する [1]。多関節弾性物体では、図 1 のように弾性物体要素はお互いが繋がっているように見せる必要があるため、a, b の値はこの条件を満たすように与えなければならない。提案手法で a, b は、 θ_1, θ_2 が 0 のときの dz に依存するとして計算した。これは、ボーンが曲がらずに、直線になったときにうまく変形する条件を想定している。境界要素法は線形近似したシミュレーションを行っているので、弾性物体の変形は線形となるため、 $a, b \propto dz$ として与える。

4. シミュレーション結果

実験では、図 3 左のような 5 つの弾性物体要素を用いて、図 3 右のモデルを構築した。頂点数は 1978 である。図 4 は、提案手法を用いてシミュレーションした結果である。実験では、以前提案した多関節弾性物体モデル [1] と同様な、伸縮や曲げの変形が可能であった。

このときの計算速度は、1 フレームあたり 1.17msec (CPU Pentium4 650 3.4GHz) であった。以前のモデルでリアルタイム処理 (1 秒間に 60 フレーム) 可能な頂点数が 2000 程度であったのに対して、提案モデルでは 28257 頂点計算可能である。また、計算に必要なメモリ (式 (4) の A' で必要となるメモリ) は、理論値では以前のモデルが 9.1MB であるのに対し、提案モデルでは 0.5MB となる。

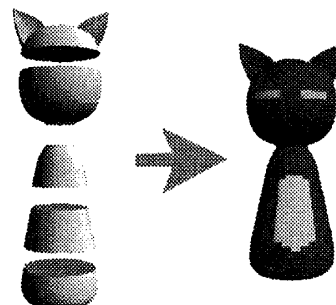


図3 多関節弾性物体モデル

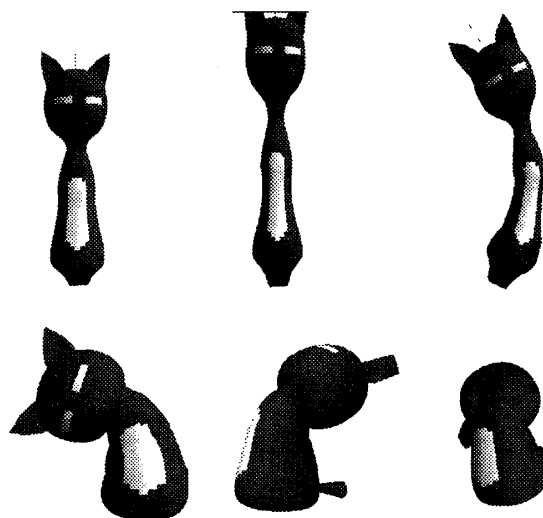


図4 図3のモデルをシミュレーションした結果

5. むすび

複数の弾性物体を接合することで、関節構造を持たせた多関節弾性物体モデルを、さらに計算コストと使用メモリの削減を可能にしたモデルを提案した。実験結果から、以前のモデルの 10 倍以上の計算速度、および、使用メモリを 10 分の 1 以下に削減可能であった。

しかし、曲げた場合の接合面の箇所における変形については、以前のモデルの方が滑らかである。今後は、より滑らかな変形ができるモデルについて検討していく。

参考文献

- [1] 渡辺隆史, 大谷淳, 棚沢順, 徳永幸生, "複数弾性物体を用いた対話操作が可能な弾性物体キャラクターモデルの構築法の検討", 情報処理学会研究報告, 2006-CG-122, 2006.02.
- [2] D.L. James, D.K. Pai, "ARTDEFO: Accurate real time deformable objects," ACM SIGGRAPH99, pp.65-72, 1999.