

位相操作を用いた 3 次元形状モデルのデータ圧縮法

増 田 宏[†] 大 瀧 竜太郎^{††} 青 野 雅 樹^{††}

3D CAD の普及により、製品設計などで大規模な 3 次元形状モデルが多数作成されるようになっていく。今後、製品データの標準化が進むと、その膨大なデータ量がデータ交換などにおいて深刻な問題になってくると考えられる。そこで、3D モデルを情報損失なく少ないデータ量で表現することが必要となってくるが、三角形メッシュに限定した場合を除けば、これまでそのような方法は提案されてこなかった。本論文では、3D CAD で用いられるワイヤフレーム、多様体サーフェス、多様体ソリッドを対象として、形状データを圧縮する方法について述べる。CAD で利用される形状モデルには様々な位相表現が用いられるが、これらはオイラー操作と呼ばれる可逆な位相操作を施すことで、単純な位相構造に変換することができる。ここでは、面の穴や空洞などを含む複雑な位相の形状モデルを位相的に三角形分割したモデルに変換し、その際用いた位相操作とワイヤフレーム/三角形分割モデルを符号化して圧縮することを考える。復元の際は、復号化した形状モデルに逆位相操作を順次施していくことで元の位相を復元する。符号化されたデータでは、形状モデルの位相と幾何が近傍のデータからの相対的な量で記述されているので、形状に現れる位相や幾何の繰返しパターンがエントロピー符号化を用いることで効率的に除去され、大きな圧縮率を得ることができる。

Coding of 3D Geometric Models Using Topological Operations

HIROSHI MASUDA,[†] RYUTAROU OHBUCHI^{††} and MASAKI AONO^{††}

We propose a compression method for 3D geometric models, which include wireframe, manifold surface, and manifold solid models. Compression methods have been proposed so far for encoding triangular meshes. However, they cannot encode 3D models created by 3D CAD systems, since they could include a wide variety of topological structure, such as holes, strut edges, isolated edges in faces, and cavities in solids. Our method first simplifies the topology of a given 3D model by using a sequence of Euler operators. The simplified model is encoded together with the Euler operators sequence. Using our method, an arbitrary surface or solid model can be encoded after it is converted into triangular meshes. To decode reverse Euler operators are applied to triangular meshes to obtain the original 3D model. Finally the encoded data are compressed using entropy coding. Since topological or geometric patterns that appear repeatedly in a shape have coherency. This redundancy is effectively removed by using entropy coding. The experimental result shows that our method achieves excellent compression ratios.

1. はじめに

製造業などでは、設計やプロセスプランニング、解析などに 3 次元形状モデル (3D モデル) が広く利用されるようになってきた。設計・生産情報のデジタル化が進むに従って、非常に多数の 3D モデルデータを記憶装置に保存したり、ネットワークを通じた高速なデータ交換が必要になってくる。しかし、製造業で一般に利用されている境界表現の 3D モデルでは、特にソリッドモデルの場合に、データ量が非常に大きくなるとい

う問題がある。そのため、今後、設計生産のグローバル化や ISO/STEP 等の設計データ標準化などによりデータ交換が頻繁に行われるようになれば、データ量の問題は深刻になってくると考えられる。

このような問題を解決する 1 つの方法は、3D モデルを情報損失なく、できる限り少ないデータ量で表現することである。画像や音声に関してはそれぞれの性質をうまく用いた独自の圧縮法が提案されており高い圧縮率が実現できているが、3D モデルに適した圧縮法に関しては、三角形メッシュに対応した幾何圧縮 (geometry compression) の研究^{1),2)}はあるものの、CAD で広く用いられている 3D モデルに対するデータ圧縮については、その重要性にもかかわらず、これまで提案されてこなかった。

[†] 東京大学大学院工学系研究科

School of Engineering, The University of Tokyo

^{††} 日本アイ・ビー・エム株式会社東京基礎研究所

Tokyo Research Laboratory, IBM Research

一般に、3D モデルは、頂点、稜線、面などの隣接関係を示す位相データと、座標値や曲線・曲面の方程式を示す幾何データとから構成されている。しかし、従来の表現方法は、データ転送のように少ないデータ量が望ましい場合には適しているとはいえない。たとえば、ソリッドモデルの位相表現に用いられている winged-edge 構造は形状操作には適しているものの、冗長な情報も含んでおり、データ転送や保存に適した表現とはいえない。また、幾何データに関しても、座標値の性質を考慮することでデータ量を減らせることが期待できる。

本研究では、3D モデルをできる限り少ないデータ量で表現する方法について考察し、形状データの圧縮・復元システムを試作することでデータ量の評価を行う。ここでは、通常の CAD システムで用いられる形状モデルとして、ワイヤフレームおよび多様体サーフェスと多様体ソリッドを考え、位相データと座標値データを効率的に圧縮するための方法を示す。本研究の貢献は、3D モデルに可逆な位相操作を施して圧縮しやすい位相に変換し、変換された 3D モデルと逆向きの位相操作の両方を符号化することで広い範囲の 3D モデルに対応できる圧縮方法を提案したことと、3D モデルデータ圧縮システムの実装を行い評価を行ったことである。本論文では、まず 2 章で基本方針と処理の概略について述べ、以降の節でデータ圧縮の手順について詳述する。3 章では位相変換、4 章で位相データと幾何や属性データとの対応付けの方法、5 章で位相データの圧縮方法、6 章では座標データの圧縮法について述べる。7 章では実験結果を示し、最後にまとめと展望について述べる。

2. 3次元形状モデルの圧縮

2.1 形状データ表現への要求

3D モデルは、グラフ構造を表現した位相データと、数値の集合である幾何データから構成される。CAD で用いられる形状モデルの圧縮表現を考える際に考慮すべき点として、以下のことがあげられる。

(1) 情報の損失がないこと

CAD システムによって計算誤差の評価方法が異なるため、位相データを安定に再現するためには、幾何計算をせずに元の隣接関係が完全に復元されるべきである。また、幾何データに関しては、表示目的の圧縮には浮動小数値を量子化して整数近似する方法が有効であるが¹⁾、幾何計算をとまなう場合にはわずかな誤差により集合演算操作などの破綻の原因になるので、忠実に再現されることが必要である。

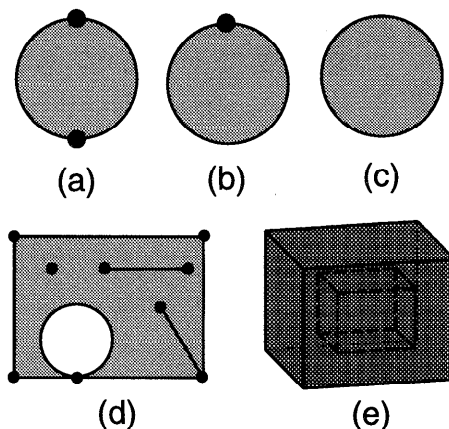


図 1 3D CAD で用いられる形状モデルの位相表現の例
Fig. 1 Examples of topological representation used by 3D CAD.

(2) 多様な位相表現への対応

CAD のモデルではかならずしも三角形メッシュのような「きれいな」位相だけでなく、図 1 に示すような 2, 1, 0 角形 (a~c)、単純でないループから構成される面 (d)、複数のシェル (ここでは連結なサーフェスの集合をシェルと呼ぶ) から構成される空洞のあるソリッドなどを許すことが多い。また、3D CAD では目的に応じて、ソリッド、サーフェス、ワイヤフレームのいずれもが広く利用されているので、これらすべてに対応できることが望ましい。

(3) 外部参照の保持

面や稜線に対して公差や解析情報など様々な属性が付加されることがあるので、それらの参照関係が維持される必要がある。

2.2 データ圧縮の手順

CAD の 3D モデルにおいて、三角形メッシュの場合のような幾何圧縮技術が提案されていない理由の 1 つは、(2) で示したような位相の多様性にある。例外的な位相の存在を認めればそれだけ豊かな表現形式を採用しなければならず、また位相的な前提知識を利用しにくくなるという問題がある。そこで、本研究では位相変形操作 (オイラー操作) に基づいた位相変換を用いることにより、位相的に単純なデータに変換してから、位相と幾何の符号化を行う方式を提案する。符号化したデータは、冗長性を除去するために、さらにエントロピー符号化を用いて圧縮する。エントロピー符号化には辞書式圧縮やハフマン符号化、算術符号化などがあり、いずれを用いてもよいが、本研究ではエントロピー符号化として辞書式圧縮⁶⁾を用いるものとする。

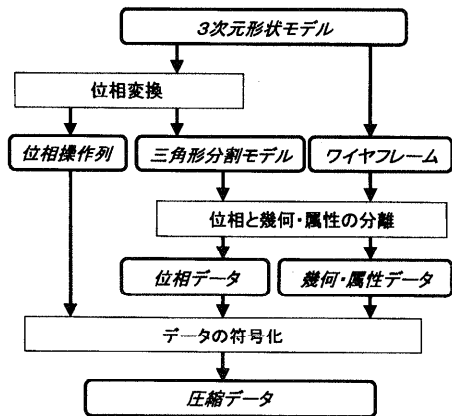


図2 3次元形状モデルのデータ圧縮の手順

Fig. 2 A process for coding 3D geometric models.

図2にデータ圧縮の手順を示す。まず、与えられた3Dモデルに位相変換を施して圧縮に適した位相構造に変換し、このとき位相変換に用いた位相操作を記録する。ここでは、ソリッドとサーフェスは三角形分割モデルに変換する。次に、位相データと幾何または属性データを異なる手法によってそれぞれ符号化し、これと位相変換で用いた位相操作列を符号化したものを組み合わせて3Dモデルの圧縮データとする。復号のときはこれらを逆にたどり、位相データの復元、幾何や属性データの復元、それらを結合した3Dモデルの復元、逆向きの位相変換による元の3Dモデルの復元、という手順で行う。各手順で行われる処理を以下に簡単に説明しておく。

(1) 位相変換：まず、3Dモデルが複数の部品から構成されているときには、サーフェスの集合とワイヤフレームの集合に分離する。さらにサーフェスに対して位相操作を施して面の穴や0, 1, 2角形などをなくした後、すべての面が位相的に三角形となるように分割された三角形分割モデルに変換する。そのうえで、三角形分割モデルとワイヤフレームモデルに対して符号化を行う。なお、ソリッドモデルは、ソリッドであることを示すフラグの付いた閉じたサーフェスとして処理する。復号のために、逆向きの位相操作に必要なデータは符号化して圧縮データの一部として保持する。

(2) 位相データと幾何や属性データの分離および結合：本手法では、幾何データと位相データを別々に符号化するが、復号時に両者の対応が再現できる必要がある。また、位相要素に対して属性が付加されている場合には、位相要素とその属性との対応も保持されなければならない。ここでは稜線を枝とする展開木を用いてすべての頂点を順序付けすることで、位相データ

と幾何や属性データとを対応付ける。展開木は符号化し、圧縮データの一部として保持する。

(3) 位相データの圧縮：位相変換の結果生成されたワイヤフレームモデルと三角形分割モデルに対して位相データの符号化を行う。これらの位相は単純であるため大きな圧縮率を見込むことができる。ここでは、ワイヤフレームの圧縮には両端の頂点番号リストの差分のエントロピー符号化、三角形分割モデルの圧縮には Taubin ら²⁾の提案した符号化法を用いる。

(4) 幾何データの圧縮：座標値の表現についてのみ考えるが、同様の考え方は、曲線や曲面の表現についても適用可能である。ここでは、幾何データは倍精度の64ビットの数値列であると考え、近傍の点との差分をとったうえでエントロピー符号化を用い、冗長性を除去する。

3. 位相変換

3.1 位相操作

位相変換では、任意の3Dモデルをワイヤフレームと三角形分割モデルの集合に変換する。そのような変換は、以下に示す位相操作を順次施していくことで実現できる。これらの操作はいずれも可逆であり、逆向きの操作を施すことで元の位相構造が復元できる。

(1) 複数シェルの分離：ソリッドモデルが空洞を持ち、複数のシェルから構成されるときは、それらを分離する(図3(1))。この操作によって、ソリッドモデルを向き付けされた閉じたサーフェスと見なせる。

(2) サーフェスの稜線分割：3Dモデルによっては、図3(2)のような端点があた1つの稜線や、端点を持たないループが許容されているので、これらがあれば頂点を追加して稜線を分割する。また、複数の稜線が両端点を共有するときには一方の稜線を分割し、始点と終点から稜線が一意に決まるようにする。

(3) 単一ループ面への変換：面に空洞があるときには、面は2個以上のループで構成される。このときは図3(3)に示すように、同じ頂点を一度だけ通るただ1つのループから構成される面の集合に分割する。操作は、ループの併合、面の分割の順で稜線を追加していくことで実現される。

(4) 三角形分割：多角形を三角形に分割する。ここでは、座標値を考慮する必要はなく、位相的に三角形分割できていれば十分である。そこで、図3(4)のように1点を共有する三角形の集合に反時計回りに順次分割していく。

3.2 位相操作の符号化

位相操作の対象となる位相要素は、次節で説明する

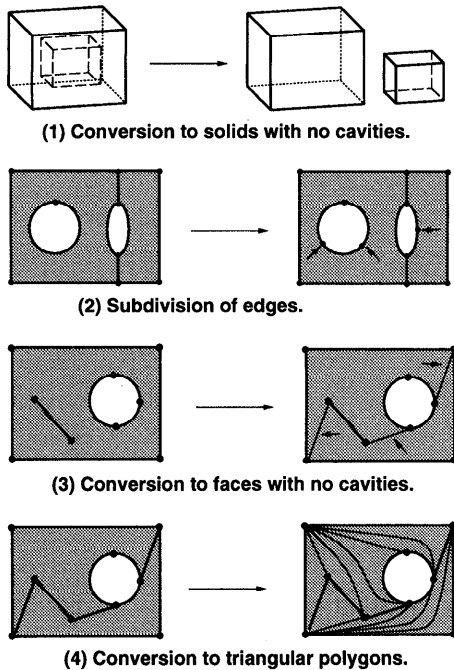


図3 位相変換で用いられる位相操作
Fig. 3 Operations for topological transformation.

方法で頂点、稜線、面に付けられたID番号を用いて記述する。これらをID番号でソートし、差分をとり、さらに差分をエントロピー符号化することで圧縮データの一部とする。

(1) シェルの包含関係：シェル間の包含関係が復元できるように外郭シェルとそれに含まれるシェルを保持しておく。シェルは、そのシェルを構成する任意の一点のIDによって表現する。

(2) 除去要素：圧縮の際に追加した頂点と稜線を復元の際に除去するために、除去すべき要素IDを保持する。三角形分割では3.1節の(4)で述べたように反時計回りに稜線を追加しているので、個々の稜線IDは保持せず、最初の稜線IDと向き、追加した稜線の個数を保持しておく。稜線の向きは、両端点の頂点IDの小さい方が始点のとき0、それ以外は1というように1ビットで表現できる。

(3) ループ復元：位相操作(3)では外郭ループと穴ループを併合するが、その際、どのループが外郭ループであるかという情報が失われる。それを保持するために、外郭ループ上のある1つの稜線IDと、稜線の向きがループの向きと一致するかを示す1ビットのフラグを保持する。

(4) ソリッド復元：閉じたサーフェスとソリッドを区別するための1ビットのフラグを保持する。

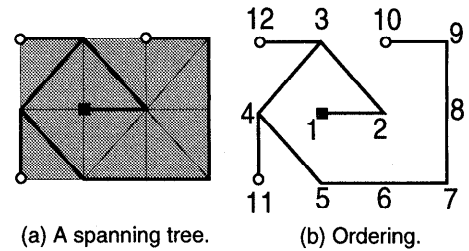


図4 展開木による頂点の順序付け
Fig. 4 Ordering vertices using a spanning tree.

4. 位相データと幾何データの分離と結合

位相データと幾何データは、圧縮の際に分離して別々に符号化するが、復元の際には再び結合しなければならない。そこで、展開木を用いることで位相データに含まれる頂点、稜線、面に一意の1次元順序を保つようにID番号を付け、幾何データはこのID番号の順序に従って列挙することで位相と幾何の対応関係を保つことにする。

まず、頂点の順序付けの方法を説明する。図4(a)は、与えられた多面体モデルを覆う展開木の例を示しており、■は開始頂点、○は端点である。頂点での番号付けは深さ優先で行い、開始頂点から順次、ループを作らないように、連結な稜線の端点を選び、選んだ点の順に番号を振っていく。なお、サーフェスとソリッドの場合には、5.2節で述べる位相圧縮のために、次の頂点として時計回りに最も近い稜線の端点を選ぶ。図4(b)は、この展開木の頂点を順序付けをしたものである。

稜線と面の順序付けは、頂点IDに基づいて行うことができる。稜線の場合には、両端の頂点IDで $[v_i, v_j]$ ($v_i < v_j$) と表し、2つの稜線 $[v_{i1}, v_{j1}]$, $[v_{i2}, v_{j2}]$ の大小関係を、 $(v_{i1} < v_{i2})$ または $(v_{i1} = v_{i2}, v_{j1} < v_{j2})$ のときに $[v_{i1}, v_{j1}] < [v_{i2}, v_{j2}]$ と定義して稜線をソートし、その順序を稜線IDとする。面の場合も、ループ上の最小の頂点IDを始点とし、頂点IDを反時計回りに並べた $[v_1, v_2, \dots]$ として記述できるので同様にソートして番号を付けることができる。曲線や曲面が含まれる3Dモデルの場合にはこのIDを用いて位相データと幾何データを対応付ける。

展開木は符号化して圧縮データの一部として保持し、復号時に位相データと幾何や属性データとの対応が保てるようにする。展開木の符号化では、各分岐点に2つのフラグを付ける。1つは、展開木を頂点IDの順にたどっていくときにその分岐点を始点とするたどられていない枝があるかを示すフラグ、もう1つは○で

示すような端点を示すフラグである。ここで枝別れを記号‘(’、終端を記号‘)’で表すと、図4の展開木は、

1 2 3 (4 (5 6 7 8 9 10) 11) 12)

と記述できる。連続するIDの個数を用いて書き換えれば、結局この展開木は3(1(6)1)1)と符号化することができる。

5. 位相データの圧縮

5.1 ワイヤフレームモデルの符号化

すでに展開木が符号化されているので、それに含まれないワイヤのみを符号化する。展開木に含まれないすべてのワイヤを4章で述べたように両端の頂点番号で表現してソートした $[s_i, e_i]$ ($i = 1, \dots, n$) を求め、さらにこれらの差分をとった $[s_i - s_{i-1}, e_i - s_i]$ ($i = 1, \dots, n$, ただし $s_0 = 0$ とする) によりワイヤを表現する。これらに辞書式圧縮などのエントロピー符号化を用いることで、大きな圧縮率が達成できる。高い圧縮率が達成できるのは、位相的に近傍の頂点には近いIDが割り振られるために、(1) 差分をとることで桁数が小さくなる、(2) 同一の差分値が多くなる (0, 1, 2 がほとんどである)、(3) 差分をとることで局所的な位相が符号化され、類似の位相構造には類似の符号列が対応することが多い、という性質があるためである。

5.2 三角形分割モデルの符号化

本手法では、サーフェスとソリッドを三角形分割モデルに変換しているため、既存の三角形メッシュ圧縮手法を利用することができる。三角形メッシュの圧縮方法についてはこれまでいくつか提案されているが、ここでは比較的圧縮率の高い Taubin の方法²⁾を用いる。この手法を、図4を例にとって簡単に説明する。このモデルに対して図5のように、まず展開木に沿ってモデルを切断し、次に、2辺が展開木にある三角形 (Δ で示す) を出発点とする破線矢印で示すような三角形の帯を考える。三角形の帯は、矢印の向きに、前の三角形に接する稜線を底辺として、次の三角形が左右どちらの稜線に接するかを示す1ビットのフラグの列によって符号化できるので、長い帯が生成できれば非常に高い圧縮率が実現できる。このような符号化をすべての帯について行い、これらの帯を連結するための情報を保持することで形状モデルの位相構造を保持する。復元のときは、三角形の帯を展開木に沿って縫い合わせることで元の位相構造に戻すことができる。

6. 座標データの圧縮

ここでは、頂点座標の圧縮の問題を考える。座標

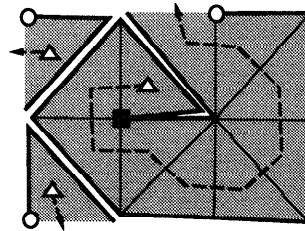


図5 三角形分割モデルの圧縮

Fig. 5 A method for coding triangular meshes.

データは (x, y, z) の浮動小数点値を前章の手法で付けた頂点番号の順に列挙することで記述できる。

座標値列の圧縮においては、 x , y , z それぞれの1次元数値列を考え、線形予測符号化^{2),3)}や残差予測ベクトル量子化⁴⁾に基づいて冗長性除去をした後、ハフマン符号化などでエントロピー圧縮をする方法が提案されている。この方法は、浮動小数点が16~24ビット程度の整数に正規化でき、かつ座標値列が滑らかに変化する場合には有効である。しかしながら、CAD用途の形状モデルでは幾何計算に64ビットの倍精度を用いることが多く、しかも表示用モデルのように曲面を多数の三角形で近似しないのが一般的である。制御点が密である曲線や曲面データの圧縮には線形予測符号化が有効であると考えられるが、頂点座標の圧縮に関しては、我々の実験の結果、必ずしも有効ではないことが分かった。

そこで、座標値圧縮には、主として繰返しの冗長性を除去するために、差分値のエントロピー符号化を用いるものとする。座標値を近傍の頂点からの差分ベクトルとして記述することで類似の数値が生じやすくなり、大きな圧縮を見込むことができる。番号 p の頂点の差分値としては、連結な頂点のうち p を超えない最大の番号 q の頂点座標との差をとる。たとえば、図4の頂点12では頂点3との差分ベクトルがとられる。ただし、頂点 p と q の x 座標をそれぞれ x_p , x_q とするとき、差分 $x_p - x_q$ の指数部が x_p の指数部よりも大きくなる場合には桁落ちが生じるため、 $(x_p - x_q) + x_q$ は x_p に等しくならない。そこで1ビットのフラグを付けて、桁落ちを生じる場合には x_p をそのまま保持することにする。また逆に $x_p - x_q$ の指数部が小さい場合には下位ビットを保持するのは無駄なので、必要のないビットは切り捨てる。さらに、IEEE標準の倍精度では指数部に11ビット使うが、実際に用いる指数の範囲はこれよりはるかに少ないので、最大の指数値との差をとることで、指数部のビット数を減らすことにする。

7. 実験結果

形状モデルのデータ圧縮システムの実装においては、サーフェス、ソリッド、ワイヤフレームの形状表現と位相操作を統一的に扱うために、radial-edge 構造の非多様体形状モデラ⁵⁾を用いた。例題としては図6の形状1, 2を用いた。これらをソリッドモデルとして扱った場合と、面情報を無視してワイヤフレームモデルとして扱った場合のそれぞれについて表1に示す。圧縮結果は、位相データだけの圧縮と、位相データと幾何データを合わせた場合の圧縮について示した。従来形式として、位相データに関しては稜線を端点の頂点番号で、面を稜線番号で記述した典型的な ASCII データ、幾何データとしては倍精度の数値を 64 ビットで記述したバイナリファイルを用いた。また、圧縮の結果、従来形式のデータの何%になったかを括弧内に示した。さらに、従来形式のデータを gzip (LZ77 と呼ばれる辞書式圧縮のアルゴリズムに工夫を加えて実装したプログラム。詳細は文献6) 参照) で圧縮し

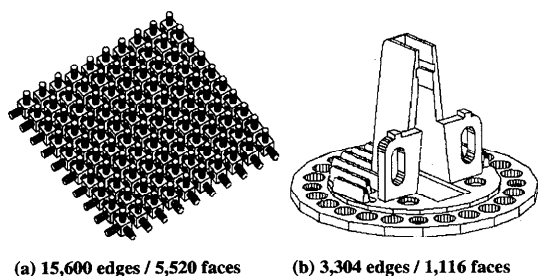


図6 データ圧縮の評価に用いた例題
Fig. 6 Examples of 3D geometric models.

表1 図6の形状モデルの圧縮結果 (括弧内は従来形式との比較)
Table 1 Compression ratios of 3D geometric models shown in Fig. 6.

	3D モデル	従来形式	LZ77 (gzip)	本手法
a の 圧 縮 率	ワイヤフレーム	153.89 KB	54.75 KB	0.23 KB
	位相のみ	(100.00%)	(35.58%)	(0.15%)
	ワイヤフレーム	403.49 KB	87.86 KB	3.66 KB
	全体	(100.00%)	(21.78%)	(0.91%)
	ソリッド	495.52 KB	186.59 KB	11.43 KB
	位相のみ	(100.00%)	(37.66%)	(0.31%)
b の 圧 縮 率	ソリッド	745.12 KB	219.70 KB	14.86 KB
	全体	(100.00%)	(29.49%)	(2.00%)
	ワイヤフレーム	29.72 KB	12.67 KB	0.66 KB
	位相のみ	(100.00%)	(42.63%)	(2.22%)
	ワイヤフレーム	82.48 KB	23.81 KB	6.73 KB
	全体	(100.00%)	(28.87%)	(8.16%)
	ソリッド	93.35 KB	38.80 KB	4.21 KB
	位相のみ	(100.00%)	(41.56%)	(4.51%)
	ソリッド	146.11 KB	49.94 KB	10.28 KB
	全体	(100.00%)	(34.18%)	(7.03%)

たデータも比較の対象として示した。

これにより、いずれの場合も、本手法により大幅なデータ圧縮が実現できることが分かる。なお、位相圧縮において、位相操作の符号化データが位相データ全体に占める割合は (a) が 8%, (b) が 14% である。本手法では、同一の位相パターンには同一のビット列が割り当てられる傾向があるため、同一の位相パターンが多く現れる形状1に関しては特に大きな圧縮率が得られることが分かる。

8. 結 論

本論文では、CAD で用いられる 3 次元形状モデルであるワイヤフレーム、多様体サーフェス、多様体ソリッドを対象として、位相データと幾何データを圧縮する方法について述べた。本手法は、位相操作を用いて、3 次元形状モデルを単純な位相構造に変換することで広い範囲の形状モデルに適用可能であるという特徴を持つ。また、本手法を例題の 3D モデルに対して適用し、従来の表現形式を汎用の圧縮プログラムを用いて圧縮した場合に比べて、十分大きな圧縮率が得られることを示した。本手法の位相と幾何の表現では、位相と幾何を近傍のデータからの相対的な量で記述するため、形状に現れる位相や幾何の繰返しパターン検索をビット列の冗長性除去の問題に帰着させることができ、エントロピー符号化を用いることで大きな圧縮率を得ることができる。

本論文においては幾何データとして頂点の座標値のみを扱ったが、実際の 3D モデルの圧縮を行うためには、NURBS やトリム曲面などの圧縮も考慮する必要がある。これらに関しては、今後拡張していく予定である。また、3D モデルによっては頂点列に関してもなんらかの予測が有効となることが考えられ、その場合には予測値との差分をとることでさらに圧縮率を向上させることが期待できる。CAD の 3D モデルに適した座標値の予測法についてもさらに研究が必要である。さらに、本研究では本手法を実際に使われている多数の形状モデルに適用して評価するということを行わなかったが、広い範囲の形状に適用して本手法の圧縮率傾向を評価することも必要である。

参 考 文 献

- 1) Deering, M.: Geometry Compression, *ACM SIGGRAPH'95*, pp.13-20 (1995).
- 2) Taubin, G. and Rossignac, J.: Geometry Compression Through Topological Surgery, IBM Research Report, RC-20340, pp.1-22 (1996).

- 3) 大淵竜太郎, 増田 宏, 青野雅樹: 3次元形状モデルの座標データの圧縮, 平成9年後期情報処理学会全国大会論文集, Vol.4, pp.317-318 (1997).
- 4) Horne, C. (Ed.): Coding of Moving Pictures and Audio, ISO/IEC JTC1/SC29/WG11 N1821r, pp.46-54 (1997).
- 5) 増田 宏: 非多様体形状モデルのための形状操作とその応用に関する研究, 東京大学博士論文 (1996).
- 6) Nelson, M. and Gally, J.-L.: データ圧縮ハンドブック第2版, トッパン (1996).

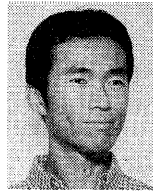
(平成10年2月10日受付)

(平成10年5月8日採録)



増田 宏 (正会員)

1961年生. 1985年東京大学工学部精密機械工学科卒業. 1987年同大学院工学系研究科修士課程修了. 同年日本アイ・ビー・エム(株)入社, 東京基礎研究所に勤務. 1998年より東京大学大学院工学系研究科助教授. 3次元形状処理, CG, 設計支援システムに関する研究に従事. 工学博士. IEEE, 人工知能学会, 精密工学会, 日本造船学会各会員.



大淵竜太郎 (正会員)

1981年上智大学理工学部電気電子工学科卒業. 1983年電気通信大学計算機科学科修士課程修了. 1994年 University of North Carolina at Chapel Hill Computer Science Department より Ph.D. 取得. 1994年より日本アイ・ビー・エム(株)東京基礎研究所所属. インタラクティブコンピュータグラフィクス全般, 特に augmented reality とその医療への応用に興味を持つ. ACM, IEEE, 日本ソフトウェア科学会各会員.



青野 雅樹 (正会員)

1957年生. 1981年東京大学理学部情報科学科卒業. 1984年同大学理学系大学院研究科修士課程修了. 同年日本アイ・ビー・エム(株)入社. 1990年より米国ニューヨーク州レンセラー総合工科大学計算機科学科に留学し, 1994年に Ph.D. 取得. 同年より日本アイ・ビー・エム(株)東京基礎研究所所属. ACM, IEEE 各会員.