

ネットワークモデルを用いた 空間分割モデルの形状圧縮

藤村真生 小堀研一 久津輪敏郎
大阪工業大学

1. はじめに

筆者らは空間分割モデルを用いてCADシステムを構築するためのデータ構造として、ネットワークモデルを提案した⁽¹⁾。ネットワークモデルはデータ数を少なく抑えてかつ立体集合演算を高速に行なうことが可能な空間分割モデル⁽²⁾の一種である。

本稿ではこのネットワークモデルによって表現された形状を更に圧縮し、空間分割モデルの形状表現に必要なデータ数を大幅に削減する手法について説明する。また、従来から空間分割モデルの形状を圧縮する手法として一般に用いられるオクトツリーモデル⁽³⁾を取り上げ、ボクセルモデルにより表現された形状を圧縮する比較実験を行った。この際に必要となる処理時間と圧縮後のデータ数をそれぞれ比較し、新しい手法の有効性を示して考察する。

2. ネットワークモデル

空間分割モデルは、3次元形状をボクセルと呼ばれる微小な立方体の集合として表現する。ネットワークモデルは形状表面のボクセルのみを形状のデータとして保持するモデルである。この方法により、ボクセルモデルと比較してデータ数を1%以下に抑えることができ、形状演算の際の計算回数や描画時間等を削減することができる。しかし、形状表面のボクセルのみを離散的な状態で保持しただけでは、形状に対する任意の点の内外判定ができず立体集合

演算に応用することは不可能である。そこで図1に示すように形状表面に存在するボクセルを双方向のポインタによって接続し、形状表面においてネットワークを構成することによりこの問題を解決した。

ネットワークモデルは形状内部の情報を持たないので扱うボクセルの数が少なく、かつ総てのボクセルが均一のサイズであるから、形状の内部のデータを保持するボクセルモデルや異なる大きさのボクセルを持つオクトリーモデルと比較して形状に回転・移動を施す幾何演算に適した形状モデルとなる。

3. ネットワークモデルの形状圧縮

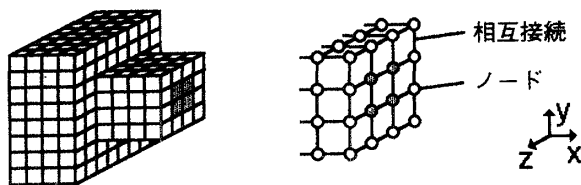
ネットワークモデルの形状圧縮は、「面領域の圧縮」と「稜線部の圧縮」という2つの手順により実現される。これらの手順について詳しく述べる。

3.1. 面領域の圧縮

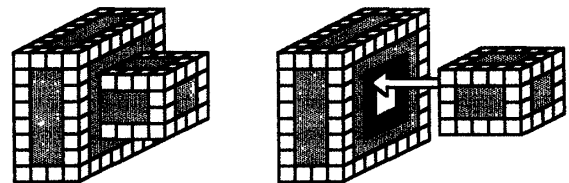
図1に網掛けで示した部分のボクセルは、 yz 平面に平行な同一平面上に存在している。図1aの形状からこのように同一平面上に存在するボクセル群を面領域としてまとめると、図2aに網掛けで示した面群が抽出される。この面群と面領域に属さなかったボクセル群とによって同一の形状を少ないデータ数で表現することが可能となる。これを面領域の圧縮と呼び、次にその手順を示す。

処理1：面領域として認識可能なボクセルを、接続しているポインタの方向成分がすべて同一平面上にあるか否かで判定する。この判定により面領域と認識されたボクセル群をグルーピングする。

処理2：グルーピングされたボクセル群と同一の平面上に存在しながら、接続しているポインタの方



a. 形状 b. 右側先端部の接続の様子
図1 ネットワークモデル



a. 面領域 b. 突起部分の接続の様子
図2 面領域の圧縮

Data Compression of spatial partitioning representations using NETWORK model.

Masao Fujimura, Ken-ichi Kobori, Toshiro Kutsuwa

Osaka Institute of Technology

16-1 Omiya, Asahi, Osaka-City 535, Japan

向成分が平面上にないものを、形状の外側から見て右回りに辿りループを生成する。これにより、面領域の外郭部分のループと図2bに黒塗りで示した面領域の内部に存在する内側のループが生成される。

処理3：面を表現するノードを生成し、処理2で生成したループをチェーン構造を用いて保持する。

処理4：処理1でグルーピングされたボクセル群を削除する。この時削除されるボクセルに接続されていたポイントの接続先を、外郭ループの外側のボクセル群は外郭のループに、内側のループの内側に存在するボクセル群は内側のループに変更する。

3.2. 稜線部の圧縮

図2のボクセル群から稜線部として認識可能なものをまとめる。これにより図3に太線で示した稜線群と面領域の圧縮により生成した面群、および面領域

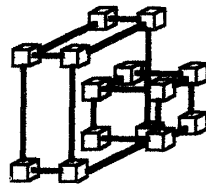


図3 稜線部

・稜線部のどちらにも属さなかったボクセル群とによって図1の形状を表現することが可能となる。これを稜線部の圧縮と呼び、次にその手順を示す。

処理1：面領域として抽出されなかったボクセル群から2つの面ループに接続するものを抽出し、面ループの組み合わせによってグルーピングする。

処理2：稜線を表現するノードを生成する。グルーピングされたボクセルの数を稜線の長さとし、またポイントの方向を稜線の方向として保持する。

処理3：グルーピングされたボクセル群をすべて削除し、残ったボクセルの接続先を変更して新しく生成した稜線のノードに接続させる。

4. 実験結果および考察

ネットワークモデルによる空間分割モデルの圧縮の有効性を検証するため、オクトツリーモデルによる形状圧縮との比較実験を行った。

実験ではSGI社のIndyを用い、ボクセルモデルにより表現された形状を上記の2つの手法によって圧縮した処理時間と圧縮後のデータ数を比較した。実験に用いた形状は球、立方体、円柱、円錐、4面体、プリズムの基本形状6種類、カメラ、パソコン、電話機等の製品形状8種類、基本形状を単純に組み合わせて生成した形状6種類である。結果の平均値を

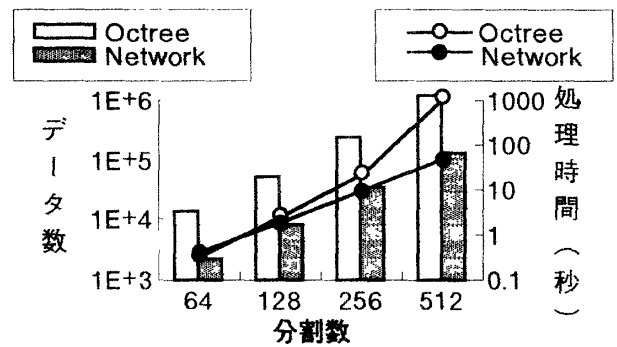


図4 実験結果

算出したものを図4のグラフに示す。

図4の折れ線グラフから、ネットワークモデルによる圧縮の処理時間は空間の1辺を512分割した場合において1分程度になる。オクトツリーモデルと比較すると分割数の低い場合には同程度であるが、分割数を上げた場合の処理時間の増加が少ない。オクトツリーモデルによる圧縮ではボクセルモデルのすべての領域にわたって形状の内外判定をする必要があり、その処理時間は分割数の3乗に比例する。一方ネットワークモデルでは形状表面のボクセルのみを検索し内外判定を行うため、分割数の2乗に比例した時間で処理を完了することが可能となる。

また、図4の棒グラフに示した結果から、ネットワークモデルによる圧縮を行うと、オクトツリーモデルと比較してデータ数を11%から17%程度に抑えることが可能となる。

5. おわりに

ネットワークモデルによる圧縮は、オクトツリーモデルと比較してデータ数は17%程度になり、また処理時間の分割数に対するオーダーも削減できることが実験により明らかとなり、空間分割モデルの圧縮方法として有効であることが明らかとなった。

参考文献

- (1) 藤村, 小堀, 久津輪: "高速演算可能なネットワーク空間分割グラフィックスモデルの提案と評価", 電子情報通信学会論文誌 (D-II), Vol. J80-D-II, No3, pp.783-790(1997)
- (2) J.Weng, N.Ahuja: "Octrees of Objects in Arbitrary Motion", Computer Vision, Graphics, and Image Processing, 39, pp.167-185 (1987).
- (3) F.Foley, A.VanDam, S.Feiner & J.Hughes: "Computer Graphic Principles and Practice Second Edition", Addison Wesley Publishing Company, pp.533-562(1990).