

オクトツリーモデルのハードウェア化

1C-5

藤村真生 小堀研一 久津輪敏郎
大阪工業大学

1. はじめに

工業デザインのためのモデリングシステムにおいて、デザイナーが操作しやすい形状モデルの一つとして空間分割モデルを用いることが考えられる。空間分割モデルを扱う上で問題になるのは、モデル精度の向上に伴う使用メモリ量と計算時間の爆発的な増加である。それを補うべくソフトウェアとデータ構造の改良による高速化の手法が提案されている¹⁾。しかし実際の工業デザインに要求される精度を実現するには至っていないのが実状である。そこで本発表では、専用のハードウェアによる計算の高速化の手法について提案し、回路のシミュレーションの結果によりその優位性を検証する。

2. オクトツリーモデル

オクトツリーモデル²⁾は3次元の形状を表現するための空間分割モデルと呼ばれるデータ構造の一種である。従来CADに多く利用されている境界表現は、頂点・稜線・面などにより形状を表現するため、デザイナーがそれらの要素について意識しながらモデリングする必要があるため、入力に問題があった。これに比較して空間分割モデルはボクセルと呼ばれる微小な直方体の集合によって形状を表現するため、デザイナーが各要素について考慮する必要が無く、より現実に近いモデリングが可能となる。

空間分割モデルを扱う上で問題となるのが計算機の記憶容量と計算に要する時間である。オクトツリーモデルは、モデリングを行なう形状に応じて大

きなボクセルで表現できる部分は大きなボクセルによって、また細かな表現が必要な部分は小さなボクセルで表現する。このため、モデリング精度を等しくした場合は空間全体を均一に細かく分割するボクセルモデルと比較して扱うデータ数を抑え、またモデリング操作の際の計算時間や画面表示にかかる時間も抑えることが可能となる。

また、空間分割モデルにおける幾何計算は3軸に平行な直方体に関するものが基本となり、かつ各々のボクセルは互いに干渉し合わない。よって各空間に対して並列的な処理が可能であるので、ハードウェア化に適していると考えられる。

3. システムの概要

本発表で提案するシステムの概要を図1に示す。本システムはオクトツリーモデルを用いたモデリングにその主眼をおき、専用のハードウェア（以後エンジンと呼ぶ）を用いて高速な計算を行なうことを目的とする。このため、エンジンはデザイナーが実際に操作するパーソナルコンピュータ（以後計算機と呼ぶ）に搭載される形で動作する。

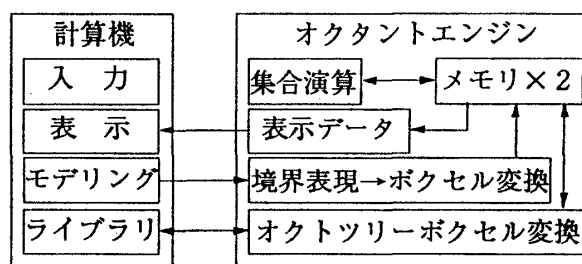


図1 システムの概要

エンジンに搭載されるメモリと集合演算部にはボクセルモデルを用い、計算機との通信ではオクトツリーモデルによるデータ圧縮を行なう。そこでエンジンには、次に示す4つの機能が要求される。

Hardware engine of Oct-tree data

Masao Fujimura, Ken-ichi Kobori, Toshiro Kutsuwa
Osaka Institute of Technology
16-1 Omiya Asahi-Ku Osaka-City Osaka 535, Japan

1. 形状データの集合演算.
2. オクトツリー・ボクセルモデルの相互変換.
3. 境界表現からボクセルモデルへの変換.
4. 形状表示用のデータの合成.

また、計算機は次に示す4つの機能を持つ.

1. デザイナーからの操作入力.
2. モデリングデータの作成.
3. 形状の表示.
4. 形状ライブラリの構築.

本システムは工業デザインのためのシステムである。そこでモデリングで扱う空間全体を一辺約50cmの立方体とし、それに対して一辺1mmの立方体までを表現できる精度を目標とする。

4. エンジンの開発

システムの核となるエンジンの開発にはハードウェア記述言語 (HDL Hardware Description Language) を用いた。本研究で用いる HDL は、NTT の開発した SFL³⁾ である。まず SFL によってエンジンの演算部などの動作記述を行ない、この記述から同じく NTT の開発した CAD システム PARTHENON による論理合成・シミュレーション等を行なって LSI を製作する。

5. シミュレーション

オクタントエンジンによる高速化の有効性を検証するため、試作した回路によるコンピュータシミュレーション実験を行なった。試作した回路はエンジンの演算部の一部であるが、本エンジンの処理の中で最も重要な部分である。

試作した回路は境界表現からボクセルモデルへの変換の際に必要となる、ボクセルと無限平面との交差判定処理、および3次元クリッピング処理である。前者の処理の流れは次のようになっている。

- 1) まず空間全体を表わす大きなキューブの8つの頂点と無限平面との符号付きの距離を算出する。
- 2) 空間を8分割する。
- 3) 1の判定において8つの距離が総て同符号でなければ、分割されたキューブ各々に対して1から2の処

理を再帰的に繰り返す。ただしこの際、各々の頂点と平面との距離は分割する前のキューブが持っていた距離の値から線形補間すればよい。

4) 1の判定において8つの距離が総て同符号だった場合、分割されたキューブ各々に対して2の処理のみを再帰的に繰り返す。

この処理は単独で無限平面による形状の削除等の処理が行なえるほか、平面とボクセルとの交差判定を行なう場合の処理の一部として重要な役割を持つ。

6. 実験結果

試作した回路をシミュレーションした結果と、同じ処理を SGI 社の Indy (R4400, 150MIPS) を用いてソフトウェアで行なった場合との比較結果を表1に示す。比較の際には空間全体を x,y,z 方向に各々512分割したオクトツリーデータを用いた。

表1 実験結果

処理	Indy	エンジン
無限平面との交差判定	13sec	0.27sec
3次元クリッピング	210nsec	4.0nsec

7. おわりに

シミュレーション結果から、オクタントエンジンによる交差判定処理およびクリッピング処理の高速化については有効であると判断できる。今後他の回路についてもシミュレーションを行ない、検証を行なう。

参考文献

- 1) 米川, 中山, 小堀, 久津輪: 空間分割モデルを用いた形状モデラ, 第10回ヒューマン・インタフェース・シンポジウム論文集, pp.177-182, 1994
- 2) Foley, van Dam, Feiner, Hughes: Computer Graphics Principles And Practice Second Edition, Addison-Wesley Publishing Company, pp.533-562, 1990.
- 3) NTT データ通信株式会社: PARTHENON User's Manual, 1990