

J-25

分散処理による3次元モデリング処理の高速化

A Fast Voxel Modeling Using Distributed Processing

中村 徳裕† 西尾孝治† 小堀 研一†
 Norihiro Nakamura Koji Nishio Ken-ichi Kobori

1. まえがき

現在, CAD, CG分野では一般的に境界表現モデルが用いられている。しかし, 境界表現モデルではユーザは形状要素の接続情報(位相構造)を意識する必要があり, 操作に熟練を要する。また, モデリングの際に頻繁に行われる形状同士の集合演算の処理負荷が大きく, 複雑な処理では実時間処理が困難になる。

一方, 空間分割モデルの一種であるボクセルモデルは, データ構造が単純である, 立体集合演算が容易に行なえる等の特徴があり, 形状モデラとしても提案されている^[1]。しかし, 解像度が低いと物体表面の凹凸が大きく, 解像度を高くするとデータ量が空間の解像度の3乗に比例して大きくなる。そのため, 表示負荷や形状操作時の処理負荷が増加し, リアルタイムに処理することが困難となる。また, 高解像度になるに従い, 必要なメモリ量も膨大となり, 汎用的なパーソナルコンピュータでは扱えない。

そこで, 本研究では精度の高いボクセルモデルを扱うために, ネットワークで接続された複数のコンピュータを用いた分散処理モデリング手法を提案する。

2. 3Dモデリングシステムの概要

上記のボクセルモデルの問題解決のためにクライアント1台, サーバ複数台の分散処理モデリングシステムを構築する。その際, 高解像度のボクセルデータの変形結果をリアルタイムに表示するためにネットワークでの送受信時間の短縮, 形状操作を行う際の処理時間の短縮を実現する。

なお, データの信頼性を保つためにネットワークプロトコルとしてTCP/IPを用いる。

本研究では, 各サーバにボクセル空間を均等に分割したデータを持たせ, それぞれのサーバが自身の管理する空間のみに処理を行うクライアントサーバシステムを構築する。これにより, 解像度の3乗に比例して処理負荷が大きくなるボクセルモデルの問題を解決する。

2.1 ネットワーク構成

本稿では, 図2.1のようにクライアントを1台, サーバを複数台とするネットワークを構築した。

クライアントは各サーバ間でやり取りされるデータのルーティングを行い, 各サーバは分割されたボクセル空間内の形状操作の実処理を行う。

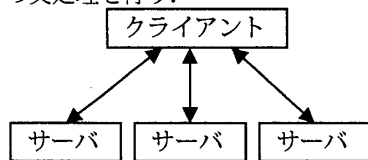


図2.1 ネットワーク構成

2.2 機能

モデリングの機能として切断, 盛り付け(削りだし), 削除の3つの機能を用意して, 手法の有効性を検証する。

2.2.1 切断

切断面が複数のサーバの管理する領域にまたがって存在する場合, サーバの境界部分で切断面を分割し, 各サーバが自身の管理する領域のみに並列して処理を行う。

2.2.2 盛り付け, 削りだし

あらかじめ用意された基本形状と, モデリング対象形状との集合演算によって行う。選択された基本形状が複数のサーバの管理する領域にまたがって存在する場合, 各サーバが自身の管理する領域に含まれる部分のみに対して並列して処理を行う。

2.2.3 削除

削除は, 形状内部を黒画素, 外部を白画素とした場合に削除したい閉領域を白画素で塗りつぶすことで行う。選択された削除対象領域が複数のサーバの管理する領域にまたがって存在する場合, 各サーバが削除対象の領域のうち自身の管理する部分のみに処理を行い, 削除対象領域が他のサーバにまたがる場合にはまたがっているサーバにそれを知らせる。この処理を再帰的に繰り返すことによって, 選択領域を削除する。以下にサーバの境界をA-B, 初期シードを図2.2(a)のC1とした場合の処理手順について示す。また, 実際には処理は3次元で行っているが簡単のため2次元の図で示す。

- (1) 初期シードC1をもとにスキャンラインシードフィルアルゴリズム^[2]を用いて塗りつぶしを行う。
- (2) 対象領域が複数のサーバにまたがっている場合, 図2.2(a), (b)のように境界に接した領域を塗りつぶす際に, 図2.2(b)の他のサーバへ送るシードC4があるかを判定し, シードがあれば図2.2(c)のようにクライアントを経由して対象のサーバにそのシードを送る。

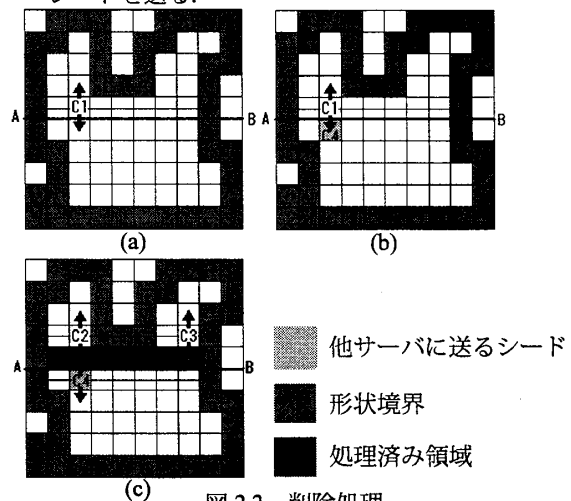


図2.2 削除処理

† 大阪工業大学情報科学部情報科学研究科

2.2 ボクセル空間の分割法

本システムでは、前処理としてボクセル空間を適切な形でサーバの台数で分割し、各サーバにそれぞれ分割されたボクセル空間を割りあてる。以下にサーバを S1~S4 までの 4 台、オクタント分割を 1 回とした場合の分割手順を示す。また、実際には分割は 3 次元で行っているが簡単のため 2 次元の図で示す。

- (1) 図 2.3 (a) のように、空間全体をあらかじめ指定した回数だけ分割する。
- (2) 分割した Q1~Q4 を、式 (1) を用いてサーバ台数だけ等分割する。Q1 の場合、図 2.3 (b) のように分割が行われる。また、式 (1) の解が複数存在する場合、 $|N_{xy}-N_{yz}|$, $|N_{yz}-N_{zx}|$, $|N_{zx}-N_{xy}|$ の 3 つができるだけ 0 に近い組み合わせを分割に用いる。これは、分割を行った際に、個々の領域ができるだけ偏りの少ない直方体になるようにするためである。
- (3) クオドラントを分割した空間を各サーバに分配する処理を、全てのクオドラントについてそれぞれ行う。例えば S1 の場合、Q1~Q4 のそれぞれのクオドラントの S1 の部分のみが分配される。また、この際に各クオドラントとサーバの管理領域との対応をもった 4 分木を作り、クライアント側で保存しておく。
- (4) ボクセルデータを受け取ったサーバ側でも同様に各クオドラントとサーバの管理領域との対応をもった 4 分木を作って置く。

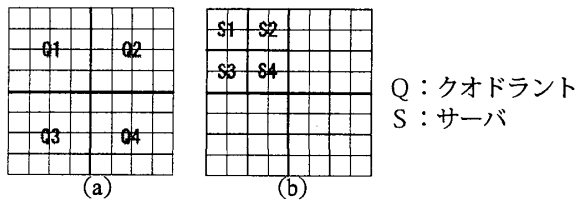


図 2.3 ボクセル空間の分割法

$$D = (N_{xy}+1) \cdot (N_{yz}+1) \cdot (N_{zx}+1) \quad \dots \text{式 (1)}$$

D : 分割数 (サーバ台数)

N_{xy} : x-y 平面に平行に分割する回数

N_{yz} : y-z 平面に平行に分割する回数

N_{zx} : z-x 平面に平行に分割する回数

この分割法では、各サーバはそれぞれ各クオドラントに対して 1 つだけ処理領域を持つことになる。このことにより、ノードアドレス法^[3]を用いてクオドラントを特定することで、サーバの担当領域が離れて複数存在していても処理対象となっているかを高速に判断することができる。また、クオドラントに分割してから同一のアルゴリズムで分配を行うため、単純に分割する場合に比べて各サーバの担当領域の処理の偏りを軽減することができる。

3. 実験と考察

3.1 処理時間の計測

最も処理負荷が大きい削除処理を行った場合の処理時間の計測を行った。5 種類の形状での結果を図 3.1 に示す。分散処理のサーバには Pentium IV 1.6GHz, OS は WindowsXP を用い、100Base のイーサネットによって接続した。また、ボクセル空間の一辺の分割数を 2^{LEVEL} とした場合の LEVEL を 9 とし、それぞれの形状を完全に削除するまでの時間を計測した。

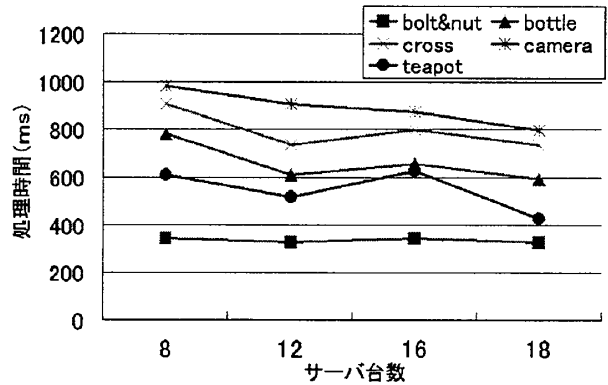


図 3.1 削除処理時間

図 3.1 からわかるように 5 形状は全て 1 秒以下で削除することができた。この実験では形状全体を削除する時間を計測したが、通常モデリングする際には形状の一部に対して操作を行う。そのため、十分に実時間で処理することができると考えられる。また、基本的にサーバが増加するにつれて処理速度が上昇しているため、サーバの増加によってより高解像度のボクセルデータにも対応することができると考えられる。このことより、本手法は有効であると考えられる。

3.2 モデル作成例

本研究で構築したモデリングシステムを使用して実際に作成した製品形状を図 3.2 に示す。

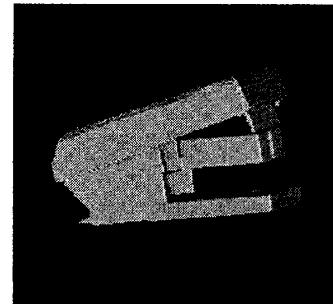


図 3.2 モデル作成例

4. あとがき

本研究ではボクセルモデルを用いることで、ユーザが位相情報を意識することなく直感的にデザイン作業を行うことが可能な 3 次元モデリング処理の提案を行った。従来、処理負荷が大きくパーソナルコンピュータではリアルタイムでの処理が困難であった高解像度のボクセルデータを、クライアントサーバシステムを用いた分散処理により処理負荷を軽減することで、対話的な形状操作を可能にした。

今後としては、より広帯域のネットワークを導入して本システムの実用化の可能性を検証していきたい。

参考文献

- [1] 手島, 図子, 西尾, 小堀: ボクセルモデルを用いた 3 次元モデリングシステムの構築, 設計工学会誌, Vol.37, No.8, (2002)
- [2] David F Rogers, 山口富士夫 訳: “実践コンピュータグラフィックス”, 日刊工業社, pp.102, pp.108, (1987)
- [3] 國井, 藤代: ソフトウェア工学ハンドブック, オーム社, pp.465-468, (1986)