

リアルタイム集合演算表示に関する研究

2V-7

床井浩平

和歌山大学経済学部

1. はじめに

CSG (Constructive Solid Geometry) はソリッドモデラにおける形状データの表現法の一つだが、境界表現を用いるソリッドモデラにおいても、会話的に複雑な形状を作成するのに有効な手段として、ユーザインタフェースの部分に使われている。

後者の場合、プリミティブの位置決めの後、集合演算処理を実行し、結果を画面表示などにより確認する。その際、結果が意図したものと異なれば、集合演算処理を取り消して、以前の形状を回復する必要がある。一般に複雑な形状に対する集合演算処理には時間がかかるため、このサイクルでは良好な操作性を得られない。

そこで形状データ自体には変更を加えず、隠面消去処理の際にブール結合関係を参照して可視面を決定する方法が考えられる。これを高速に実行できれば、集合演算の結果を見ながらプリミティブの位置決めを行うようなことも可能になる。

本研究はこれにスキャンライン法に基づくアルゴリズムを用い、ソフトウェアによりリアルタイムに集合演算表示を行うことを目的とする。

2. スキャンライン法による集合演算表示

集合演算表示の方法には、視線探索法を拡張したもの、スキャンライン法を拡張したもの[1]、Zバッファ法を拡張したものなどがある。特にZバッファ法を高速に実行するハードウェアを用いれば、ステンシルバッファを利用してリアルタイムに集合演算表示が実行できる。しかし、近年のマイクロプロセッサの性能向上により、ソフトウェアでもこれを実現できる可能性がでてきた。この点でスキャンライン法に基づく方法は、最も効率がよいと考えられる。

A Study of Real-Time CSG Display
Kohe Tokoi
Wakayama University

スキャンライン法による集合演算表示は、多面体の隠面消去処理におけるサンプルスパンでの奥行き比較による可視面の判定の代わりに、奥行き方向に1次元の集合演算処理を実行することによって実現できる[1]。この1次元の集合演算処理を効率よく実行するには、サンプルスパンと重なる面分を、視点から近い順に並べ替えておく必要がある。

3. CSGの記述と集合演算アルゴリズム

CSGの記述には木構造を用いることが多い。この方法はプリミティブを会話的に追加していく方式の形状モデリング作業との親和性がよい反面、複雑な形状になると木が深くなり、木の探索自体に時間がかかるようになる。集合演算表示では可視面決定のたびに集合演算処理を繰り返すため、これは処理速度に大きく影響する。

そこで本研究ではCSGの記述にパターンマトリクス[2]を用い、サンプルスパン単位の1次元の集合演算処理をインクリメンタル計算により実行する方法[4]、およびCSGの記述に別段のデータ構造を用いず、プリミティブの正負および登録順序を用いる方法[5]を試みた。

3.1. パターンマトリクス

パターンマトリクスはプリミティブのブール結合を積和形のブール演算式で表し、マトリクス状の表を用いて記述する方法である。差集合演算の表現には負のプリミティブを用いる。木構造で記述された形状は、記号処理により容易にパターンマトリクスに変換できる。

和集合演算は隠面消去処理において面分の交差を処理すれば実現できるので、実際には積の項（セグメント）についてのみ処理すればよい。これは以下のインクリメンタル計算で実現できる。

個々のセグメントに対して、それに属している正のプリミティブの数（リファレンスカウント）をあらかじめ求めておく。サンプルスパンに重なる面分について、視点に近いものから順に調べ、表ならリファレンスカウントを1増し、裏なら1減ずる。この結果、リファレンスカウントが0になった面分が可視面となる。負のプリミティブについては、あらかじめ面分の表裏を反転しておくだけでよい。

この方法は高速だが、一つのプリミティブが複数のセグメントに属している場合は、その全てについてインクリメンタル計算を行う必要があり、形状が複雑になるにつれてやや処理が遅くなる。

3.2. プリミティブの正負のみによる方法

積集合演算を省いてしまえば、プリミティブの正負および登録順序を用いて、木構造に似た形の形状記述が可能になる。この場合の集合演算処理は以下のアルゴリズムで実行できる。

サンプルスパンに重なる面分について、視点に近いものから順に取り出し、表なら優先順位表に登録し、裏なら優先順位表から削除する。この後、優先順位表の先頭に正のプリミティブの表の面分が存在すれば、取り出した面分が可視面となる。

このアルゴリズムはブール結合関係の記述を参照することなく可視面が決定できるため、処理時間が形状の複雑さに依存しない。

4. 集合演算処理回数の削減

個々のプリミティブの表現に WED (Winged-Edge Data) 構造を用いることによって、稜線や面分の接続関係を参照して集合演算処理を省略できる。

あるサンプルスパンにおいて可視の面分に属する稜線が、次のサンプルスパンとの境界になっているとき、その稜線を共有するもう一方の面分が次のサンプルスパンにおいて可視となる[1]。

また、ある走査線の処理が終了した後、次の走査線において進入あるいは退出する稜線が存在せず、かつ稜線の順序にも変化がなければ、処理が終了した走査線上の面分の交差を含まないサンプルスパン

における可視面を、次の走査線の対応するサンプルスパンにおける可視面とすることができる。

このほか、通常のスキャンライン法による多面体の隠面消去処理では、形状データ中に面分の交差を含まないことが明らかな場合、サンプルスパン内の交差の有無の検査を省き、処理を大幅に高速化できる。集合演算表示を行う場合はこの仮定を利用できないが、プリミティブが自己交差面を持たないとするなら、画面上でプリミティブが重ならない部分について上記の検査を省略できる。画面上でのプリミティブの重なりを検出にも、インクリメンタル計算による方法が使用できる[3]。

5. まとめ

本稿で述べたアルゴリズムの処理時間は、面分の数に対してほぼ比例し、ブール結合の記述の複雑さにはあまり影響されない。また表示解像度に対しても、画像生成の際の集合演算処理の実行回数が面分の交差のない部分においては画面の解像度に依存せず、交差の部分でも走査線数にのみ依存することから、全体でもほぼ走査線数にのみ依存する。これは通常隠面消去処理と同等のオーダである。

またこのアルゴリズムは半透明表示も実現できるため、負のプリミティブを半透明表示することで切断面の確認や位置決めを行いやすくすることも可能である。したがって本アルゴリズムは、形状モデリングの際のユーザインタフェースに適している。

参考文献

- [1] Atherton, P. R.: A Scan-Line Hidden Surface Removal Procedure for Constructive Solid Geometry, Computer Graphics, Vol. 17, No. 3, pp. 73-82 (1983)
- [2] 木下: 形状モデリングを核としたCAD/CAMシステムTIPS-1, PIXEL, No. 24, pp. 70-81 (1984)
- [3] 床井, 北橋: 凸な立体の集合演算によって定義された形状のスキャンライン法による陰影画像生成, 情報処理学会論文誌, Vol. 30, No. 1, pp. 81-90 (1989)
- [4] 床井, 北橋: スキャンライン法による多面体の集合演算表示の高速化, 情報処理学会論文誌, Vol. 34, No. 11, pp. 2412-2420 (1993)
- [5] 床井: リアルタイムソリッドモデラに関する研究, 情報処理学会研究会報告, Vol. 96, No. 77 (96-CG-81), pp. 13-18 (1996)