2K-5

バッファ法による視線探索法について

豊島正剛* 磯部俊夫** 末松俊二**
*電気通信大学 **航空宇宙技術研究所

1．はじめに

多面体（６０００－１００００平面程度、以下モデル）を処理する場合、高速性を重視してZバッファ法やスキャンライン法を採用する事は多いが、視線探索法を採用した場合の実行速度や実行サイズ等について実用性を報告する。

本手法で視線探索法をベースとする理由としては、第一にアルゴリズムの単純性、拡張性等により既存の計算機システムで短期間に効果的なシステムを実現可能である事、第二にZバッファ法等の手法と異なり透視変換が不要である事、等があげられる。

2．仕様

本手法では高速化実現のためにバッファ法と外接領域法を併用する。バッファ法は隠面処理（可視面決定）と影処理のために、外接領域法は影処理と反射処理のために使用する。

基本仕様は以下の通りである。

A．モデルは多面体のみ、構成平面（以下、物体）は、
　３角形と４角形のみとする。
B．影処理を実現する。但し点光源１個。
C．反射処理を実現する。但し反射の深度は３回とし、
　屈折処理は扱わない。
D．アンチエイリアシングは基本的に実現しない。
　但しマッピング処理でのみ実現する。

3．実験

本手法の評価にあたり、以下の条件で実行した。

A．計算機は富士通汎用計算機Ｍ７８０を使用する。
B．使用言語はＦＯＲＴＲＡＮ。
　プログラム上では倍精度実数を使用する。
C．解像度は　５１２＊４８０画素。
D．物体数は約６０００個。モデルと背景のみ。
E．表示時間は全てＣＰＵ時間であり、
　これにはコンパイル時間も含まれる。

3．1　バッファ法についての検証

領域を前処理し、物体毎に隠面処理を実行するバッファ法の性能を、視線探索法で一般に知られている以下の構成により評価した。但し効果は隠面処理のみ（影・反射処理は無し）とする。

a．単純構成（高速化の対処なし、ピクセル毎に処理）
b．領域テスト構成（各物体の画面上での存在領域を前処理し、ピクセルが領域内に含まれる物体について交点計算を実行する）
c．ＢＯＸ構成（１ＢＯＸを物体数１００個以下で構成し、交差ＢＯＸに属する物体に対して交点計算を実行する）

（画像については、後述するサンプル２の隠面処理のみ実現した場合と同じである。）

結果としては、aの場合の約１／８００程度、b・c場合の約１／１０程度の処理時間を実現する（ちなみにバッファ法の場合で約３０秒で処理完了）。

高速性実現の理由としては、第一にコヒーレンス効果が発揮されている事、第二にデータ量は多いものの画面上での局所存在性（各物体は小さく、集中している）が顕著である事、が考えられる。このバッファ法を影処理にも適用した場合、同一構成で約５０秒で完了する。（但し、この評価にあたり空間分割法を実現する余裕がなかったので機会があれば報告したい。）

本手法により、隠面処理・影処理での高速化は実現可能である。但し隠面処理については他の手法でも実現可能ではあるが、今回は視線探索法で統一した。

バッファ法を影処理にも適用する事により、簡単なシャドウポリゴンを形成する。これはシーン（舞台）全体をサポートする必要もなく主人公であるモデル（特に物体数の多いもの）に絞って実行する事により高速化を実現した。影バッファの範囲外のみ外接領域法を実現すればよい。

3．2　バッファの存否についての検証

本手法では複数のバッファを使用し、更に（仮想）スクリーンを準備してスクリーンを可動式にしている。バッファ法を採用した場合、処理速度での効果は期待できるが、問題は実行サイズである。

今回は対策として、Ｚバッファ（ＸＹＺ座標）ではなくＴバッファ（距離）を使用し、更に可動式スクリーンではあるが配列（テーブル）を準備せず、ピクセル座標・スクリーン座標共にその都度再計算させている。しかし（交点計算自体の処理時間の割合が高いので）実行時間・表示画像にその影響はみられない。

更に、Ｔバッファを使用する場合でも、スクリーン分割により一時配列の軽減を実行した場合、コヒーレンス効果減少の影響もなく（４分割で時間にして約１．０３倍）効果的な省容量化になる。本手法では一括・分割処理を隠面・影処理と使い分けてある。

特にバッファ法と言っても、全て実現する必要はなく、分割処理も可能で、プログラミング言語によっては容量も小さくなるので、システムにより実現度を選択すればよい。

4．実行

本手法により幾つかの作品を制作しその実行結果・状況を示す。実行条件は実験とほぼ同じで、実現度は影処理と反射処理を実現する。

4．1　サンプル１

人工衛星、（写真１）
多面体数　２４０個。
モデルのみ、背景無しの構成。
全物体について反射面指定をする。
１枚分の計算時間は以下の通りとなった。
　A．影無、反射無　　約１０秒
　B．影有、反射有　　約２４秒
影有・反射有で１８０枚（アニメ６秒分）の画像データを計算した場合、約７０分　との結果になった。

4．2　サンプル２

ヘリコプター、（写真２）
多面体数　６１００個。
モデル及び背景の構成。
全物体について反射面指定をする。
１枚分の計算時間の場合、
　A．影無、反射無　　約２９秒
　B．影有、反射無　　約４９秒
　C．影有、反射有　約３２０秒
となる。

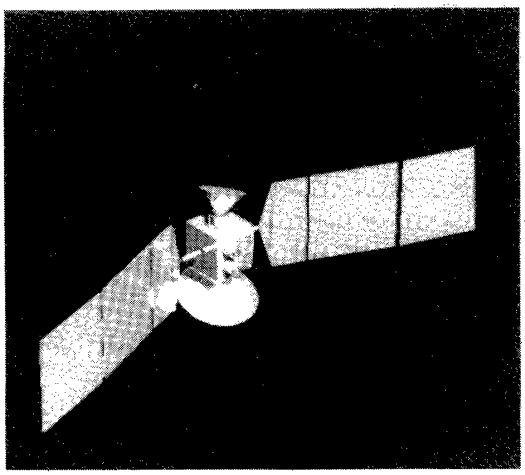
4．3　サンプル３

ヘリコプター、（写真３）
多面体数　６２００個。
モデル及び背景の構成。
影処理のみ実現（反射処理無し）。
背景についてマッピング処理有り。
１枚分の計算時間の場合、
　A．影無、　　約１２０秒
　B．影有、　　約１６０秒
となる。
又、影処理有りで９０枚のデータを計算した場合、
約１７８分　との結果となった。

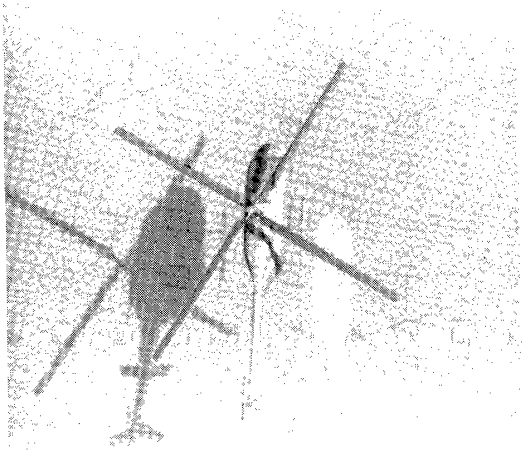
4．4　補足

バッファ法に限らず、背景等の表示領域の大きな物体が存在するとＣＰＵ時間がかかる。サンプル２の場合もモデルのみならば約１５秒程度で完了するので、考慮する必要がある。更にアンチエイリアシングを考慮したマッピングを実現するならば時間がかかる事は明白である。

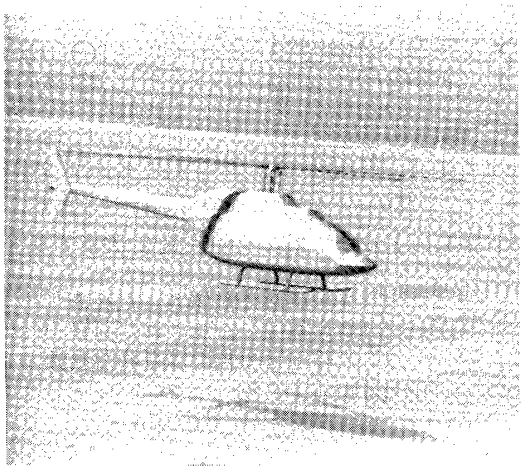
Ray tracing by Zbuffer method
Seigou Toyoshima :University of Electro Communications
Toshio Isobe,
Shunji Suematsu : National Aerospace Laboratory



サンプル１　写真１



サンプル２　写真２



サンプル３　写真３

５．まとめ

５．１　考察

本手法により、隠面処理での高速化は実現可能である。反射処理と分けて考え、各々の処理に合わせた方法を採用する事により、汎用機で或る程度の処理が実現可能である。

領域テスト自体も簡単なもので特別な前処理は不要であり、少しの変更で多面体以外の物体にも実現可能である。但し十分に効果を発揮するためには物体データの形状に関して配慮が必要である。細長い物体については領域分割よりも形状分割の方が簡単であり、この場合物体数が増加しても隠面処理の効率は向上する。但し反射処理への影響も考慮する必要がある。

５．２　課題

本手法の課題として、

第一に、反射処理を実現して１枚３２０秒程度ではまだまだ速いとは言えない。但し今回は外接領域法と言っても階層化まで実現していないので、反射光追跡部分の改良で１５０秒程度の実現を検討中である。

第二に、バッファ法に限らず、物体数の多いモデルに対する影処理が完全でなく、隣接する物体間の判定が微妙で、光源との距離に対して各物体が小さい場合に影響が大きい。オーサンプリングでも不十分であり、物体数によっては隣接物体データを指定するのも大変である。特に本テーマの一つである物体数の多いモデルを処理する場合には解決する必要のある問題である。

第三に、実行サイズ（７ＭＢ程度）である。これは高速化実現のために使用するバッファと物体データが原因である。対策としては、物体データについては物体数を限定すればよい。バッファについてプログラム構成の分割処理等が考えられるがアニメ化を前提にするならば、画像データだけでも相当なもの（１枚約０．７３ＭＢ）になるので余分な一時ファィルは避けたい。いずれにせよ、省容量化を実現してからでなければマッピング処理等の膨大なデータを必要とする処理は、単純な模様程度しか実現できない。

又、特にＴＳＳ処理の場合には他のユーザへの配慮も必要である。ちなみに本レポートのデータは夜間に採取したものであり、経過時間はＣＰＵ時間の約３倍程度である。

５．３　今後の予定

第一に、Ｔバッファ情報を利用してアンチエイリアシング（５＊５分割）の実現が可能である（一部テスト済み）。これに関して隠面処理の領域テストにより簡単な視線探索法では消えてしまうような物体に対してもアンチエイリアシングで対処可能である。

第二に、今回のデータはモデルのみの構図が多く、画像的に面白味に欠けるが、最終目標のアニメ化の時には、モデルだけでなく背景等の周囲の環境も含めての高速化についても実現する予定である。

６．謝辞

本プログラムの実験にあたり、実験データを提供して戴いた航空宇宙技術研究所・宇宙研究グループの岡本氏、及び空力性能部の斉藤氏に感謝いたします。

７．参考文献

ａ．藤本　レイトレーシングの高速化技法について
ＰＩＸＥＬ　ＮＯ．３７　１９８５

ｂ．中前他　局所レイトレーシングによる陰影表示法
情報処理学会論文誌　ＶＯＬ．２７　ＮＯ．１１
１９８６

ｃ．日高他　マルチコンピュータシステムＭＣ－１における画像生成手法　グラフィクスとＣＡＤ　１８－４
１９８５